(4)

A Technical Report
Grant No. N00014-86-K-0742
September 1, 1986 - May 31, 1989

MULTIPLE ACCESS ALGORITHMS FOR A SYSTEM WITH MIXED TRAFFIC: HIGH AND LOW PRIORITY

Submitted to:

Office of Naval Research
Department of the Navy
800 N. Quincy Street
Arlington, VA 22217-5000

Submitted by:

P. Papantoni-Kazakos
Professor

Report No. UVA/525415/EE89/116
September 1988

SCHOOL OF ENGINEERING AND
APPLIED SCIENCE

DEPARTMENT OF ELECTRICAL ENGINEERING

UNIVERSITY OF VIRGINIA
CHARLOTTESVILLE, VIRGINIA 22901

A Technical Report
Grant No. N00014-86-K-0742
September 1, 1986 - May 31, 1989

MULTIPLE ACCESS ALGORITHMS FOR A SYSTEM WITH MIXED TRAFFIC: HIGH AND LOW PRIORITY

Submitted to:

Office of Naval Research
Department of the Navy
800 N. Quincy Street
Arlington, VA 22217-5000

Submitted by:

P. Papantoni-Kazakos
Professor

Department of Electrical Engineering
SCHOOL OF ENGINEERING AND APPLIED SCIENCE
UNIVERSITY OF VIRGINIA
CHARLOTTESVILLE, VIRGINIA

Report No. UVA/525415/EE89/116
September 1988

Copy No. ____

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

ADA199501

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) UVA/525415/EE89/116	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION University of Virginia Dept. of Electrical Engineering 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Office of Naval Research Resident Representative
6c. ADDRESS (City, State, and ZIP Code) Thornton Hall Charlottesville, VA 22901	7b. ADDRESS (City, State, and ZIP Code) 818 Connecticut Ave., N.W. Eighth Floor Washington, DC 20006
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Office of Naval Research 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-86-K-0742
8c. ADDRESS (City, State, and ZIP Code) 800 N. Quincy Street Arlington, VA 22217-5000	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
Multiple Access Algorithms for a System with Mixed Traffic: High and Low Priority

12 PERSONAL AUTHOR(S)
P. Papantoni-Kazakos

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical	FROM 9/1/86 TO 5/31/89	1988 September 08	40

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	

We consider a system where a single channel is shared by both high and low priority data. We assume packet transmissions from both data categories and slotted channel. In addition, we assume binary, collision versus noncollision, feedback per slot, and limited feedback sensing capabilities for all users in the system. We assume that the high priority data are generated by a well-defined finite-number user population, while we adopt the limit Poisson User model (infinitely many independent Bernoulli users) for the low priority traffic. For this system, we propose

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL R. N. Madan	22b TELEPHONE (Include Area Code) 202- 696-4217 / 22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

and analyze a transmission algorithm, which is a mixture between a deterministic tree search, for the high priority users, and a random-access algorithm, for the low priority traffic. The algorithm is stable for both traffic classes, it guarantees a strict upper bound on the delays of the high priority packets, and induces good throughput-delay characteristics for the low priority data.

TABLE OF CONTENTS

I. INTRODUCTION

When a single channel is shared by many users, for the accommodation of their transmissions, the multiple-access problem arises. This problem takes various forms, depending on the system model, and mainly on the characteristics of the users. In packet networks, when the number of the users and their packet-generating processes are well-defined and remain unchanged, multiple-access algorithms with deterministic characteristics are most appropriate; the deterministic tree search in [2] belongs in this class, which then induces better throughput-delay characteristics than those induced by the random-access class. The random-access class is most appropriate when the user population may vary, and has the advantage of inducing operations which are independent of the user population; its disadvantage is that it induces high variations in delays. The algorithms in [1], [3], [5], [6], [11], and [13] belong in the latter class.

For both well-defined and varying population user models, the existing research efforts have almost exclusively focused on uniform user populations and on absence of strict constraints imposed on transmission delays. The exceptions are the works in [7], [8], [10], and [12]. In [7] and [10], the possibility of time constraints in transmissions is considered. In [10], a uniform, infinitely large, user population is considered with a strict upper bound on transmission delays imposed, and a random-access algorithm which then inevitably induces packet losses, is adopted and studied. In [8], an interconnected two-channel system with mixed user populations is considered; a portion of the user population has the option to dynamically join either one of the two channels, which results in an acceleration effect on the delays of the latter class of users. The algorithm in [8] is random-access; it thus induces variations on delays, and cannot tolerate strict delay limitations without packet losses. In [12], a single channel but mixed traffic system is considered, where a portion of an overall infinitely large population of users generates priority data. Then, a variation of the random-access algorithm in [13] is adopted which accelerates the priority packets. The algorithm in [12] induces large variations on the delays of the priority data, however, and cannot tolerate strict delay limitations without losses.

In this paper, we take a different approach than those taken in [7], [8], [10], and [12], which combines time-constraints and nonuniform-population issues. In particular, we consider a single-channel system accommodating mixed traffic: High priority data requiring a strict upper bound on their transmission delays, and low priority data which do not impose such strict delay limitations. We require that there are not packet losses for any part of the traffic, and our objective is to meet the strict delay requirements of the high priority data with simultaneous good throughput-delay accommodation of the low priority data. The satisfaction of our objective is feasible, based on the following observation: In real systems, there is a well-defined user population which may generate high priority data. Based on this observation, we design and analyze a mixed algorithm which performs a deterministic tree search for the high priority traffic, and which has a random-access part assigned to the possibly varying population of users who generate low priority data.

The organization of the paper is as follows. In Section II, we present the system model. In Section III, we include the description of the transmission policies adopted for both the high and the low priority packets. In Section IV, we present the analysis of the overall system, in terms of throughputs and delays. In Section V, we include numerical results. In Section VI, we draw some conclusions.

II. SYSTEM MODEL

We consider a system where a fixed number of users who may generate high priority data and a large number of users who generate low priority data only, share a single common channel for their transmissions. We assume that the data from all users are in the form of packets whose lengths are identical, and we consider synchronous transmissions; that is, the channel time is divided into slots of length equal to a single packet, and packet transmissions can start only at the beginnings of slots. We adopt the idealistic case where no propagation delays exist and where the only cause of channel errors are due to collisions. In particular, we assume that a single packet transmission is always successful and that simultaneous multiple-packet transmissions (collision) cause destruction of all the involved packets which must then be retransmitted.

We consider the existence of binary, collision versus noncollision, feedback per slot; a collision slot will be denoted C, while a noncollision slot will be denoted NC. We adopt the limited sensing environment; that is, each user observes the feedback sequence continuously, only from the time he generates a packet to the time that this packet is successfully transmitted. In addition, we assume that identifiable "flags" exist in the system, which indicate the beginnings of certain slot frames; the frames and the "flags" are described in the next section. Each user observes the "flags" only from the time he generates a packet, to the time that this packet is successfully transmitted. As will be explained in the next section, the frames and their identification are necessary for securing a strict upper bound on the delays of the high priority packets, with simultaneous system stability; that is, without rejecting any data packets from the system.

We assume 2^N number of users, who may generate high priority packets. We assume that in a period of 2^{N-1} (n+2) slots, for some n such that $1 \le n \le N$, each of the 2^N users generates a high priority packet with probability p, and generates no such packet with probability 1-p. We impose the constraint that the transmission of each high priority packet cannot be delayed by more than 2^N (n+2) slots.

For the traffic generated by the low priority users in the system, together with the possible low priority traffic that may be generated by the 2^N users, we adopt the limit Poisson user model with some Poisson intensity λ packets/slot; that is, infinitely many Bernoulli independent users with total traffic intensity λ. As proven in [9], the latter model corresponds to a "worst case," in terms of throughput-delay performance of some given random access algorithm.

Our objective is to devise transmission policies for the mixed, high versus low priority, traffic, which are stable (they do not reject any packets), which satisfy the strict delay constraints for the high priority packets, and which accommodate as effectively as possible the low priority traffic.

Time will be measured in slot units, where slot t occupies the time interval [t, t+1). The feedback of slot t will be denoted x_t, where x_t=C if the slot is occupied with a collision, and where x_t=NC otherwise. It is assumed that a packet arrival in [t, t+1) observes the feedback x_t, and all the subsequent feedbacks x_{t+1},..., until it is successfully transmitted.

III. THE TRANSMISSION POLICIES

In this section, we describe the transmission policies adopted by both the high and the low priority traffics, given the models and constraints presented in Section II. We start with the

description of the policy adopted by the high priority packets, since they have to satisfy a strict delay constraint, and since the policy for the low priority packets will be designed "around" the openings allowed by the former policy.

III.1. The Policy for the High Priority Packets

Let us temporarily ignore the presence of the low priority traffic, and let us only consider the high priority packets generated by the 2^N users. Let the 2^N users be indexed from 1 to 2^N, and let this indexing be known to them all. Let the 2^N users form 2^{N-n} disjoint groups, where $1 \le n \le N$, and where the k-th group contains the users with indices from $(k-1)2^n+1$ to $k\,2^n$, $1 \le k \le 2^{N-n}$. Let n and the grouping be known to the 2^N users. Then, in the absence of low priority traffic, the high priority packets are transmitted by "examining" the 2^{N-n} groups sequencially, from group 1 to group 2^{N-n}; that is, all possible packets in group k are transmitted after those in groups 1 to (k-1) have completed successful transmission. The transmissions within each group are accommodated via the rules of some multiple-access algorithm, and the transmissions from group k start immediately after the last successful transmission from group (k-1). We point out that in the presence of low priority traffic, the transmissions from group k will not start immediately after the resolution of group (k-1). Instead, to allow openings for the low priority traffic, the transmissions from group k will start after the "worst case" resolution length for group (k-1) has been exhausted. This will be explained better, in the process of this section.

Let us first describe the multiple-access algorithm used for the successful transmission of all packets within a group containing 2^n users. The algorithm is the deterministic tree-search version of the 2-cell random access algorithm in [11]. The reason we adopt this deterministic tree-search rather than the probabilistic (random) access imposed by the algorithm in [11], is due to the strict upper bound delay constraint in conjunction with the unacceptable of rejections for the high priority packets. Indeed, no random-access algorithm can simultaneously satisfy the above two requirements. The deterministic form of the algorithm in [11] is described as follows:

> Consider the binary tree in Figure 1, with 2^n leaves. The 2^n users are placed on the latter leaves. The resolution of the tree starts with a root transmission; that is, all users who may have a packet to transmit, first attempt transmission. The number of slots needed for the tree resolution (for the successful transmission of all the packets on its leaves) is called the Collision Resolution Interval (CRI). If the feedback from the root transmission is NC, the CRI lasts one slot. If, instead, the feedback from the root transmission is C, then a collision resolution process starts with the next slot. The process works with binary subdivisions of the tree, exactly as with the Capetanakis tree algorithm [2], until the first after the root collision successful transmission. The difference here is that after each successful transmission, and before the end of the CRI, the tree-search starts from the tree root.

As compared to the Capetanakis tree algorithm, the algorithm here has the advantage that a CRI which starts with a collision ends the first time that two consecutive NC slots appear. This fact makes the ends of CRIs easy to identify for users in the limited sensing environment. In addition, as its random-access counterpart [11], the present algorithm has much higher resistance to feedback errors than that of the Capetanakis tree algorithm.

We now present a useful proposition, whose easy proof can be found in the Appendix.

Proposition 1

Given the binary tree with 2^n leaves, the longest CRI induced by the algorithm in this section corresponds to the case that each one of the 2^n users has a packet to transmit, and this largest length equals:

$$L_n^{max} \stackrel{\Delta}{=} 2^{n-1}(n+2) \qquad (1) \square$$

In view of the proposition, given 2^N users who are divided into 2^{N-n} groups, for some n such that $1 \leq n \leq N$, we visualize 2^{N-n} consecutive time "frames," each of length $2^{n-1}(n+2)$, comprising a "superframe." Then, we visualize the channel time being divided into consecutive superframes, which are in turn subdivided into consecutive frames (see Figure 2). According to the model for the high priority packets, as presented in Section II, each of the 2^N users can generate at most one packet per superframe length, with probability p. We then propose the following transmission policy for the high priority packets:

> Let us consider the superframe i in Figure 2. Then, during this superframe, all the high priority packets that were generated during the superframe immediately preceeding it, are successfully transmitted. In particular, the users in the first of the 2^{N-n} user groups transmit their packets within the first frame in superframe i, where the corresponding CRI starts with the first slot of this frame. The collision resolution process follows the rules of the algorithm in this section. Similarly, the users in the k-th user group transmit their packets within the k-th frame in superframe i, where the corresponding CRI starts with the first slot in the frame.

From the above description, and in view of the fact that the probability p is less than one, it is clear that the CRIs in each frame will be generally shorter than a frame length. Thus per frame, some sequence of consecutive slots (ending with the last slot in the frame) will be generally free, to be used for transmissions by low priority packets. (see Figure 3).

The scheme explained in this section can work in the limited sensing environment, only if the starting points of the superframes can be identified by any new high priority packet that enters the system. We thus assume that those points are identified by "flags" which may correspond to encoded messages occupying a small percentage of a slot. Then, upon generation of a high priority packet, a user starts observing the channel, until he sees the first flag. Then, he transmits his high priority packet within the superframe following the flag; in particular, within the frame of the latter superframe that corresponds to his group.

As it is clear from above, for the high priority packets, only the identification of the starting points of the superframes is necessary. For better accommodation of the low priority packets, however, it is desirable to have the beginnings of frames identifiable as well; otherwise, the low priority users will be penalized (as we will see below) by higher delays. If such identification is possible, it must be distinct from that for the beginnings of superframes, for the benefit of the high priority packets, which cannot be synchronized otherwise. Thus, if the starting points of frames can be identified, they will be by distinct from the flags codes, called "miniflags."

We conclude this subsection, by pointing out that the transmission policy we have proposed for the high priority packets, clearly guarantees a worst case strict upper bound on the per packet delay (from the time the packet is generated to the time it is successfully transmitted). This bound equals two times the length of a superframe; thus, $2^N(n+2)$ slots, which is consistent with the constraint presented in Section II. Let us also remark, that given 2^N users, the number n is selected to satisfy the upper bound on the per high priority packet, on one hand, and to maximize

the throughput of the low priority traffic, on the other hand. This is generally accomplished by the largest possible n, which still satisfies the delay constraint for the high priority packets. Indeed, as n increases, the percentage of the per frame capacity dedicated to the low priority traffic increases as well, and so then does the throughput of the low priority traffic, for any given random-access algorithm adopted for its transmission. For upper bound on the per high priority packet that is at least 2^N (N+2), n should be selected equal to N. In the latter case, all 2^N users form a single group, and frames and superframes become then identical entities.

III.2 The Policy for the Low Priority Packets

From the preceeding, it is clear that the channel capacity dedicated to transmissions of low priority packets, consists of the portions of the frames which follow the ends of high priority CRIs. If the union of those portions could be seen as a separate channel dedicated to the transmissions of the low priority packets, then the problem becomes simple: A known limited sensing random access algorithm is adopted, and the throughput-delay performace for the low priority traffic is then predictable. The issue here is: Can low priority packets identify the portions of the frames assigned to them, and how? We will provide answers to this question, when the 2-cell random access algorithm in [11] is deployed for the transmission of the low priority traffic. As established in the latter reference, this algorithm induces CRIs whose ends are easily identifiable by new packet arrivals; thus, it can be easily implemented in the limited sensing environment. In addition, this algorithm is highly insensitive to feedback errors, in the presence of the limit Poisson user model (and nonmixed traffic) its throughput is 0.43, it induces good delays, and it is matchable with the algorithm adopted for the high priority traffic; that is, the ends of CRIs for both high and low priority traffics are identified similarly by new packet arrivals.

To make our presentation clear, it is first necessary to review briefly the operations of the algorithm in [11], when nonmixed traffic, say low priority traffic only, is present in the system. Then, the algorithm generates a sequence of consecutive CRIs, such that: Let at the end of some CRI, the total length of arrival intervals which contain packet arrivals that have not yet attempted transmission be d, called the lag. Then, the next CRI resolves an arrival interval of length equal to min (d, Δ), where Δ is an algorithmic parameter and equals 2.33 for throughput maximization (for attaining throughput 0.43 in the presence of the limit Poisson user model). In the first slot of the CRI, all arrivals in the arrival interval of length min (d,Δ) transmit. If at most one arrivals are contained in the latter interval, the CRI ends and lasts one slot whose feedback is NC. Otherwise, the first slot of the CRI is a collision slot, and a collision resolution process starts with the slot following it. During the collision resolution process, each involved user utilizes a counter, whose value in slot t is denoted r_t, and where r_t = either 1 or 2. If r_t = 1, the user transmits in slot t. If r_t=2, the user withholds in slot t. The r_t values are updated as follows:

(a) If $r_t = 1$ and x_t = NC, the packet of the user is successfully transmitted in slot t.

(b) If $r_t = 1$ and x_t = C, then:

$$r_{t+1} = \begin{cases} 1, \text{ with probability } 0.5 \\ 2, \text{ with probability } 0.5 \end{cases}$$

(c) If $r_t = 2$ and x_t = NC, then $r_{t+1} = 1$

(d) If $r_t = 2$ and $x_t = C$, then $r_{t+1} = 2$

From the above recursions, it can be easily concluded that a CRI which starts with a collision, ends the first time after its beginning that two consecutive NC slots appear. Due to the latter property, a new packet arrival knows for sure that a CRI has ended, the first time after its arrival that it observes two consecutive NC slots. In the limited sensing environment, a packet arriving in [t, t+1) starts observing the channel feedbacks, beginning with x_t, and remains passive until the first after t occurence of two consecutive NC slots; after that, it can identify the ends of consecutive CRIs. In such environment, each CRI examines an arrival interval whose length is min (d, Δ), where the window of length Δ slides from left to right on the lag d, with its right edge being one slot before the starting point of the CRI (see Figure 4). Everytime the above mentioned packet arrival is not within the window of size Δ (which implies that the lag d is necessarily longer than Δ), it updates its arrival instant by adding a Δ value to it. The first time his updated arrival instant is closer than Δ+1 from the beginning of a CRI, the packet is successfully transmitted during the process of the latter CRI.

Let us now consider the system studied in this paper, as seen by low priority packets. The CRIs for the low priority packets will be identical to those of the algorithm operating with non-mixed, strictly low priority traffic, only that they will be generally interrupted by CRIs from the high priority traffic. The delays induced for the low priority packets will be generally longer than those induced when high priority packets are absent from the system. To present the operations of the low priority packets clearly, we will consider two cases: (A) The case where only flags, but not miniflags, exist in the system, and (B) the case that both flags and miniflags exist.

Case (A)

Let a low priority packet arrive at some point during superframe i. It immediately starts observing the channel feedbacks sequencially, beginning with that of the slot containing its arrival instant. Due to the absence of miniflags, the packet cannot identify the beginnings of frames within superframe i; thus, it cannot distinguish between ends of CRIs for high priority packets and same ends for low priority packets, within superframe i. Therefore, it observes passively, until the occurence of the first after its arrival flag (which identifies the beginning of superframe (i+1)). Since it knows the lengths of the frames, it can identify, from the occurence of the latter flag and on, the starting points of frames. Knowing the latter points, it can also identify the ends of CRIs for high priority packets, within each such frame, and thus the portion of each such frame which is assigned to transmissions of low priority packets. Within those frame portions only, the packet operates as it would if high priority packets were absent. Specifically, starting with the point when the first after its arrival flag occurs, the packet subtracts from the arrival axis the intervals that are occupied by CRIs for high priority packets, by adding to its arrival time their lengths, sequentially as they occur (see Figure 5). After it observes the first pair of two consecutive NC slots within the channel portions assigned to low priority transmissions, it also starts adding a length Δ to its arrival time, each time a CRI for the latter transmissions ends and the packet is not part of it. The packet is transmitted within a CRI for low priority packets, if its updated arrival instant is in distance at most Δ+1 from the beginning of the CRI.

Case (B)

Due to the existence of miniflags, and since only frames (rather than superframes) are relevant to the low priority packets, they ignore flags in this case. Let a low priority packet arrive during some frame, say frame i. Then, as in case (A), it immediately starts observing the channel feedbacks sequencially. The packet remains passive and unable to make any synchronization decisions, until one of the following two events occurs:

(a) It observes a second after its arrival pair of two consecutive NC slots, before it observes a miniflag. Since only the first CRI within each frame corresponds to high priority packets, the packet knows then for sure that the second pair of two consecutive NC slots corresponds to the end of a CRI for low priority users. It also knows that the remaining after the second pair of two consecutive NC slots channel time until a miniflag occurs, is assigned to low priority traffic. Thus, starting with the point when the second pair of two consecutive NC slots ends, the packet begins adding Δ length to its arrival time, each time a CRI for low priority traffic which does not include the packet ends. After the occurence of the first after its arrival miniflag, the packet can identify the per frame portions of the channel time assigned to low priority transmissions; thus, after the occurence of this miniflag, the packet updates its arrival time exactly as in case (A).

(b) The packet observes at most one pair of two consecutive NC slots between its arrival instant and the occurence of the first miniflag after that. The packet starts then the adaptations of its arrival time, exactly as in case (A), only after the instant when the miniflag occurs.

We note that as compared to the event (b), the occurence of event (a) generally results in *reduction of the packet delay.*

From the descriptions of the operations performed by low priority packets in cases (A) and (B), we easily conclude that, as expected, the existence of miniflags generally results in considerable improvement of the delays of the low priority packets, as compared to the case where miniflags do not exist. The penalty paid for such improvement is waste of channel capacity for the encoding of the miniflag signals. We note that the margins of the frames and superframes are predetermined and incorporated within the design specifications of the system, before its operation begins.

IV. PERFORMANCE ANALYSIS

In this section, we study the throughput-delay performance of the system, separately for the high versus the low priority packets.

IV.1. The High Priority Class

Given 2^N users who may generate high priority packets, given n, such that $1 \leq n \leq N$, given the multiple-access algorithm in this paper, each superframe consists of 2^{N-n} frames, where each frame has length $L_n^{max} = 2^{n-1}(n+2)$. Each superframe has thus length $2^{N-1}(n+2)$ then, and the maximum possible delay that a high priority packet may suffer is then $2^N(n+2)$. In addition, if each of the 2^N users generates at most one packet per superframe length, and if the probability that he generates such a packet is p, then p can take any value in the interval (0,1]. If p equals 1,

then every channel slot is occupied with high priority transmissions, and the throughput for the low priority traffic is zero. Given p, the average number of high priority packets per superframe is $2^N p$; thus, given n, the average number of high priority packets per slot is:

$$\lambda_{p,n} = \frac{2p}{n+2} \tag{2}$$

Given p and n, the number $\lambda_{p,n}$ signifies the rate of the high priority packets. If, on the other hand, the rate, λ_h, of the high priority packets is given, then, from expression (2), we conclude:

$$p = \frac{n+2}{2} \lambda_h \text{ , for } n : 1 \le n \le 2(\lambda_h^{-1} - 1) \tag{3}$$

If $\lambda_h \le \frac{2}{N+2}$, any integer n in [1,N] is acceptable. For given $\lambda_h < \frac{2}{N+2}$, we may use the "worst case" for the selection of the value p; that is, the upper bound $\frac{N+2}{2} \lambda_h$ (or, $p = \frac{N+2}{2} \lambda_h$).

We now proceed with the computation of the expected length of a CRI for high priority packets, within a frame, given p and n. This expected length is needed for the computation of delays for the high priority class, as well as for the evaluation of the throughput-delay performance of the low priority class. Let us define:

$L_{n|m_1,m_2}$: Given the multiple-access algorithm in this paper, given 2^n users, given that m_1 users are active (have a packet to transmit) in the upper half of the 2^n- leaves tree, given that m_2 users are active in the lower half of the tree, the expected number of slots needed for the resolution of the (m_1+m_2)-multiplicity contention.

$L_{n/m}$: Given the multiple-access algorithm in this paper, given 2^n users, given that m users are active, the expected number of slots needed for the resolution of the m-multiplicity contention.

L(n,p): Given 2^n users, given that each user has a packet to transmit with probability p, given the multiple-access algorithm in this paper, the expected length of a CRI.

We note that L(n,p) signifies the expected length of a CRI for high priority packets within a frame, and clearly,

$$L(n,p) = \sum_{m=0}^{2^n} \binom{2^n}{m} p^m (1-p)^{2^n-m} L_{n/m} \tag{4}$$

$$L_{n/m} = \frac{1}{\binom{2^n}{m}} \sum_{k=\max(0,\, m-2^{n-1})}^{\min(m,\, 2^{n-1})} \binom{2^{n-1}}{k} \binom{2^{n-1}}{m-k} L_{n/k,m-k} \tag{5}$$

$$L_{n/0} = L_{n/1} = 1 \tag{6}$$

The multiple-access algorithm induces the following recursive expression:

$$m_1+m_2\geq 2\ ;\ L_{n|m_1,m_2}=\begin{cases}2+L_{n-1|m_2} & ;\ \text{if } m_1=0\\ m_1-1+L_{n-1|m_1} & ;\ \text{if } m_2=0\\ m_1+L_{n-1|m_1}+L_{n-1|m_2} & ;\ \text{if } m_1\geq 1,\ m_2\geq 1\end{cases} \tag{7}$$

From expressions (4), (5), and (7), by substitution, interchange of order of summations, and some manipulations we find:

$$\text{For } n\geq 3;\ L(n,p)=2^{n-1}\left\{n+2-(n+5)q+\frac{7}{2}q^2-2\left[\frac{1}{q}-1\right]\sum_{k=2}^{n-1}q^{2^k}-\sum_{k=2}^{n-1}2^{-k}q^{2^k}\right\}$$

$$L(0,p)=1\quad ,\ L(1,p)=2q^2-4q+3 \tag{8}$$

$$L(2,p)=7q^2-14q+8$$

$$;\ \text{where, } q\underset{=}{\Delta}1-p$$

Remark: Given the multiple-access algorithm in this paper and 2^N users, given that in the beginning of a collision resolution interval each user may have a packet to transmit with probability p, an optimal tree depth, N-n, can be selected, such that it minimizes the expected length of the CRI; given depth N-n and probability p, the expected length of the CRI equals 2^{N-n} L(n,p), where L(n,p) is given by (8). A monotone sequence $1-\frac{1}{\sqrt{3}}\underset{=}{\Delta}q_1^*<q_2^*\cdots<q_N^*$ can be easily found, such that the (N-n)-th depth is optimal for p values such that, $q_{n-1}^*<1-p<q_n^*$. This property induces a dynamic tree-search, varying with the range of the p value, and is similar to that of Capetanakis' dynamic tree algorithm [2]. We point out that within the scenario of this paper, given N and p, the selected tree-depth (or equivalently the integer n) is not necessarily the optimal in the above sense. The choice of the integer n is then controlled by the delay upper bound for the high priority packets as well as by the throughput-delay performance it induces for the low priority traffic.

Delay Analysis

Given N, n, and probability p of a single high priority packet generation per superframe length 2^{N-1} (n+2), we wish to compute the per high priority packet expected delay D_h (N,n,p). Here, a simple version of the regenerative theorem appears, where the regenerative points are the starting points of the superframes. Let us define:

W_h: The cumulative expected waiting time of all the high priority packets generated within a superframe, before the next superframe starts, within which the above packets are successfully transmitted.

Z_h: The expected cumulative delay of all the high priority packets transmitted within a superframe, after the starting point of the superframe.

N_h: The expected number of high priority packets transmitted within a superframe.

Then, the application of the regenerative theroem gives:

$$D_h(N,n,p) = N_h^{-1} (W_h + Z_h) \tag{9}$$

where, clearly,

$$N_h = 2^N p \tag{10}$$

Assuming that each high priority packet is generated uniformly within a superframe, we also conclude:

$$W_h = 2^N p \frac{2^{N-1}(n+2)}{2} = 2^{2(N-1)}(n+2)p \tag{11}$$

for the computation of the expected value Z_h, we need the following quantities:

$C_{n/m_1,m_2}$: Given the present multiple-access algorithm, given the 2^n-leaves tree, given that there are m_1 active users in the upper half of the tree and m_2 active users in the lower half of the tree, the expected cumulative delay of the $m_1 + m_2$ packets during the CRI which starts from the tree root.

$C_{n/m}$: Given the present algorithm, given the 2^n leaves tree, given that there are m active users, the expected cumulative delay of the m packets during the CRI which starts with the tree root.

$C_{n/p}$: Given the present algorithm, given the 2^n-leaves tree, given a CRI which starts with the tree root, given that each tree leaf is occupied with a packet with probability p, the expected cumulative delay of all the packets transmitted during the CRI, starting from its beginning.

Initially, we have the following easy expressions:

$$C_{n/m} = \frac{1}{\begin{bmatrix} 2^n \\ m \end{bmatrix}} \sum_{k=\max(0,m-2^{n-1})}^{\min(m,\,2^{n-1})} \begin{bmatrix} 2^{n-1} \\ k \end{bmatrix} \begin{bmatrix} 2^{n-1} \\ m-k \end{bmatrix} C_{n|k,m-k} \tag{12}$$

$$C_{n|p} = \sum_{m=0}^{2^n} \begin{bmatrix} 2^n \\ m \end{bmatrix} p^m (1-p)^{2^n-m} C_{n|m} \tag{13}$$

In addition, from the algorithmic model and the consistency of the frames within a superframe, we easily find:

$$Z_h = 2^{N-n} C_{n|p} + \sum_{k=1}^{2^{N-n}-1} k\, 2^n p\, L_n^{max} =$$

$$= 2^{N-n}\, C_{n|p} + 2^{N+n-2}\,(2^{N-n}-1)(n+2)p$$

The algorithm induces the following recursive expressions, where $L_{n|m}$ is the quantity in (5):

$$m = m_1+m_2 \geq 2;\ C_{n|m_1,m_2} = \begin{cases} 2m + C_{n-1|m} & ;\ \text{if } m_1 = 0 \\ m(m-1) + C_{n-1|m} & ;\ \text{if } m_2 = 0 \\ m.m_1+m_2\, L_{n-1|m_1} + C_{n-1|m_1} + C_{n-1|m_2} & ;\ \begin{matrix}\text{if } m_1 \geq 1 \\ m_2 \geq 1\end{matrix} \end{cases} \tag{15}$$

$$C_{n|1} = 1$$

From the recursions in (15), by substitution in (12) and (13) and some manipulations, in conjunction with the expressions in (4) - (8), we finally obtain:

$$C_{0|p} = 1 - q = p$$

$$n \geq 1\ ;\ C_{n|p} = 2^{n-1}(1-q)\left\{ 2\prod_{i=0}^{n-1}(2-q^{2^i}) + \right. \tag{16}$$

$$\left. + \sum_{i=0}^{n-1}\left[L(i,p) + 2^{i+1} - (2^{i+1}-1)q - 2q^{2^i} \right] \prod_{l=i+1}^{n-1}(2-q^{2^l}) \right\}$$

; where L(n,p) is given by (8), and where,

$$q = 1-p$$

$$\prod_{l=j}^{i}(2-q^{2^l}) \triangleq 1\ ,\ \text{for } j>i \tag{17}$$

Substituting expressions (9), (10), (11), (14), and (16) in (9), we finally obtain, where the expressions in (17) hold, and where L(i,p) is given by (8):

$$D_h(N,0,p) = 2^N + 0.5$$

$$D_h(N,1,p) = (2^{N-1}-1)q^2 - (2^N+\frac{1}{2})q + 3.2^{N-1}+2\ ;\ N \geq 1$$

$$N \geq n \geq 2;\ D_h(N,n,p) = 2^{N-2}(n+2) + \prod_{i=0}^{n-1}(2-q^{2^i}) - \sum_{i=0}^{n-1} q^{2^i} \prod_{l=i+1}^{n-1}(2-q^{2^l})$$

$$+ 2^{-1}\left\{ (2^{N-n}-1)L(n,p) + \sum_{i=0}^{n-1} L(i,p) \prod_{l=i+1}^{n-1}(2-q^{2^l}) + \right.$$

$$+(1-q)\sum_{i=1}^{n} 2^i \prod_{k=i}^{n-1}(2-q^{2^k}) + q\sum_{i=1}^{n}\prod_{k=i}^{n-1}(2-q^{2^k})\Bigg\} \tag{18}$$

IV.2. The Low Priority Class

From the operation of the algorithmic system, as explained in Section III, it is not hard to see that the throughput, λ_l^*, of the low priority class, in expected number of packets per slot length, is determined by the throughput, λ^* of the 2-cell algorithm in [11], in the presence of the limit Poisson user model, in conjunction with the average portion, α, of the per frame capacity that is assigned to low priority transmissions. In particular, the following relationship holds:

$$\lambda_l^* = \alpha\,\lambda^* \tag{19}$$

As found in [11], $\lambda^* = 0.43$. On the other hand, for given n and p parameters, as defined in Section IV.1, and for L(n,p) as defined in the former section and as given by (8), we have:

$$\alpha = 1 - \frac{L(n,p)}{L_n^{max}} \tag{20}$$

; where L_n^{max} is the length of a frame and is given by (1) in Section III. Thus, given n and p, and due to expressions (1), (19), and (20), the throughput, $\lambda_l^*(n,p)$, of the low priority class is given by the following expression, for L(n,p) as in (8):

$$\lambda_l^*(n,p) = 0.43\left[1-\frac{L(n,p)}{2^{n-1}(n+2)}\right] \tag{21}$$

Substituting (8) in (21), we obtain, where q = 1-p:

$$\lambda_l^*(0,p)=0,\ \lambda_l^*(1,p) = \frac{0.86}{3}\,q(2-q)\ ,\ \lambda_l^*(2,p) = \frac{3.01}{8}\,q(2-q)$$

$$n\geq 3\ ;\ \lambda_l^*(n,p) = \frac{0.43}{n+2}\left\{(n+5)q - \frac{7}{2}q^2 + 2\left[\frac{1}{q}-1\right]\sum_{k=2}^{n-1} q^{2^k} + \sum_{k=2}^{n-1} 2^{-k}\,q^{2^k}\right\} \tag{22}$$

Remark. In a tedious but straightforward fashion, the following results can be found from the expressions in (22): (i) For every n, the difference $\lambda_l^*(n,p) - \lambda_l^*(n+1,p)$ is a monotonically increasing function of p, with a unique zero, denoted p_n^* ($0\leq p_n^*\leq 1$). (ii) The values p_n^*, $1\leq n\leq N-1$, form a monotone sequence; that is, $p_1^* > p_2^* > \ldots > p_{N-1}^*$. (iii) For p such that, $p_{n-1}^* > p > p_n^*$, the integer n induces the highest throughput for the low priority class; that is, if $p_{n-1}^* > p > p_n^*$, then $\lambda_l^*(n,p) = \max_{1\leq k\leq N} \lambda_l^*(k,p)$. If $p > p_1^*$ then $\lambda_l^*(1,p) = \max_{1\leq k\leq N} \lambda_l^*(k,p)$, while if $p < p_{N-1}^*$ then $\lambda_l^*(N,p) = \max_{1\leq k\leq N} \lambda_l^*(k,p)$. Due to the above results, we conclude that if the worst case delay, $2^N(N+2)$, is tolerable by the 2^N users who may generate high priority packets, (that is, if the delay constraints do not impose restrictions on the selection of the tree depth N-n), then, given p, the optimal n, which maximizes the throughput for the low priority class, may be

selected.

As a first approach in our quantitative studies, we will assume that the 2^N users who may generate high priority packets can all tolerate a worst case delay equal to $2^N(N+2)$. Then, we will assume given rate λ_h for the high priority class, and as discussed in Section IV.1, we will use the "worst case" p value; that is, $p = \frac{N+2}{2}\lambda_h$. For given N and λ_h, we will then select the n value which maximizes the throughput $\lambda_l^*(n,p)$.

Delay Analysis

To find the per low priority packet expected delays, we will relate the system considered in this paper, with a system where a single channel is assigned to exclusively low priority transmissions, and the 2-cell limited sensing random access algorithm in [11] is deployed.

Consider the following system, called prototype system: A single slotted channel is used by a limit Poisson user model, for packet transmissions. Binary, C versus NC, feedback per slot exists, and the limited sensing version of the algorithm in [11] is adopted. Given intensity λ of the Poisson process that generates the traffic in the system, let us then denote by D_λ the induced expected delay per packet, subject to λ lying in the stability region of the algorithmic system. Our objective is to relate D_λ to the expected per low priority packet delays induced when each of the two model cases, case A and case B, in Section III.2 are present and the intensity of the Poisson low priority traffic is λ. Towards that direction, we consider each of the two above cases, separately.

Case A Model: As explained in Section III.2, in this model case only "flags" exist. As compared to the prototype system described above, this model case induces the following variations: (a) In the case A model, a new low priority arrival must wait until the first after that occurence of a flag, to basically start observing feedbacks that are of any use to its synchronization with the operations of the algorithmic system; since a low priority packet arrives uniformly within a superframe length, this induces an initial, say "synchronization," delay per low priority packet, whose expected length equals half the length of a superframe. In contrast, a new arrival in the prototype system, starts with the feedback of the slot within which it arrives, to synchronize with the system algorithmic operations; since a packet arrives uniformly within a slot length, and since a slot-feedback is observed at the end of the slot, this induces an initial "synchronization" delay per packet whose expected length equals 0.5. (b) As compared to the prototype system, the model in case A induces additional delays per low priority packet, due to the per frame portion which is assigned to high priority transmissions. If β denotes the latter portion, then a single-slot length delay in the prototype system is translated to $(1-\beta)^{-1}$ slots delay in the case A model. This translation corresponds to the part of the per low priority packet delay which follows the "synchronization" delay.

Given N, n, and p, as defined in this section, given $\lambda_l^*(n,p)$ as in (21), given intensity $\lambda \in (0, \lambda_l^*(n,p))$ of the Poisson low priority traffic, let us denote by $D_\lambda^A(N, n, p)$ the per low priority packet expected delay induced by the system in this paper, when the case A model is present. Let D_λ be the expected per packet delay in the prototype system, and let L_n^{max} and L(n,p) be as in (1) and (8), respectively. Then, due to the variations that the case A model induces as compared to the prototype system, (as explained above), the following equation

evolves naturally.

$$D_\lambda^A(N,n,p) = 2^{N-2}(n+2) + (D_\lambda - 0.5)\left[1 - \frac{L(n,p)}{L_n^{max}}\right]^{-1} \tag{23}$$

Case B Model: As explained in Section III.2, this case model allows for "miniflags" which identify the starting points of frames. In terms of delays, this model varies from the case A model, only in the initial "synchronization" delay it induces per packet. Indeed, this delay is bounded from above by half the length of a frame. If given N, n, p, and $\lambda \in (0, \lambda_l^*(n,p))$, we denote by $D_\lambda^B(N,n,p)$ the per low priority packet expected delay induced by the system in this paper when the case B model is present, and for D_λ, $L(n,p)$, and L_n^{max} as in (23), then the following relationship evolves.

$$D_\lambda^B(N,n,p) < 2^{n-2}(n+2) + (D_\lambda - 0.5)\left[1 - \frac{L(n,p)}{L_n^{max}}\right]^{-1} \stackrel{\Delta}{=} D_\lambda^{B,u}(N,n,p) \tag{24}$$

For the case B model, we will be satisfied with the upper bound in (24), rather than a more precise calculation of the expected delay $D_\lambda^B(N,n,p)$, since the latter is too involved without providing significant additional information about the performance characteristics of the system. The upper bound in (24) suffices for "worst case" performance comparisons between the case A and the case B models; that is, comparison between the expression in (23) and the upper bound in (24) shows the minimum gain obtained, in terms of per low priority packet expected delays, when "miniflags" are allowed.

Due to the above discussion and expressions (23) and (24), we conclude that for the evaluation of the per low priority packet expected delays in the presence of each of the two, case A and case B, models, the expected per packet delay D_λ, in the prototype system remains to be computed. For given n and p, D_λ needs to be computed for λ values in $(0, \lambda_l^*(n,p))$, where $\lambda_l^*(n,p)$ is the throughput of the low priority traffic that the system induces and is given by (21). The methodology for the computation of D_λ is as that presented in [4], and we include the specifics for the limited sensing version of the 2-cell algorithm in [11], in the Appendix.

V. NUMERICAL RESULTS

Using the results in Section IV, in this section we select various values of the independent system parameters, and we subsequently evaluate throughputs and expected delays for both the low and the high priority traffics. The independent system parameters are the integer N, the rate λ_h of the high priority traffic, and the upper bound on the delays of the high priority packets. We then take the following two approaches.

Approach 1: Given N, we assume that the worst case delay upper bound, $2^N(N+2)$, is tolerable by the high priority packets. Then, given λ_h, we use the "worst case" p value, $2^{-1}(N+2)\lambda_h$, and we compute the throughput, $\lambda_l^*(n,p)$, of the low priority traffic, for values of the integer n in [1,N]. We subsequently select the n value which maximizes the throughput $\lambda_l^*(n,p)$. For this latter value, which is a design parameter, we then compute the per high and low priority packet expected delays; for the latter, we actually compute upper and lower bounds.

Approach 2: Given N, we assume that there exists some integer n' in [1,N], such that the upper bound on the tolerable per high priority packet delay is $2^N(n'+2)$, where n' is generally less than N. Given λ_h, and in view of (3) in Section IV.1, we then select $n^* = \min(n', 2(\lambda_h^{-1} - 1))$ and $p^* = 2^{-1}(n^*+2)\lambda_h$. Subsequently, we select various p values in $(0, p^*]$, and for each such value we select this integer n in $[1, n^*]$ which maximizes the throughput $\lambda_l^*(n,p)$ of the low priority traffic. For the latter selection, which is a design parameter, we then evaluate expected per packet delays, for both the high and the low priority traffics.

In Tables 1 and 2, we include numerical results for Approach 1. For various values of N and the rate λ_h of the high priority traffic, we first computed the probability $p=2^{-1}(N+2)\lambda_h$, and we then computed the quantities $L(n,p)$, $D_h(N, n, p)$, and $\lambda_l^*(n,p)$, in (8), (18), and (22), respectively, for n values in [1,N]; the latter results are included in Table 1. For each pair (N,λ_h) of values, we found the interger, n^*, which maximizes the throughput $\lambda_l^*(n,p)$ of the low priority traffic; the latter maximum throughput is then denoted $\lambda_l^*(n^*)$. For each selection of the pair, (n, λ_h), the integer n^* and the maximum throughput $\lambda_l^*(n^*)$ are marked by an asterisk, in Table 1. In Table 2, we include the n^* and $\lambda_l^*(n^*)$ values, for various N and λ_h choices, as well as the quantities $D_h(N,n,p)$ in (18), $D_\lambda^B(N, n, p)$ in (23), and $D_\lambda^{B,u}(N, n,p)$ in (24), for $p=2^{-1}(N+2)\lambda_h$ and $n=n^*$, and for various λ values; the quantities $D_h(N,n,p)$, $D_\lambda^B(N,n,p)$, and $D_\lambda^{B,u}(N,n,p)$ are then denoted $D_h(N,n^*)$, $D_\lambda^A(N,n^*)$, and $D_\lambda^{B,n}(N,n^*)$, respectively. The quantities $D_\lambda^A(N,n^*)$ and $D_\lambda^{B,u}(N,n^*)$ were computed from expressions (23) and (24), respectively, in conjunction with the values D_λ from Table A in the Appendix. From Table 1, we observe that given λ_h, the throughput $\lambda_l^*(n^*)$ first increases monotonically with N, it reaches a pick, and then it decreases monotonically as N increases. In Table 2, we include the pick values of $\lambda_l^*(n^*)$, for each λ_h value, and we mark them by an asterisk. From Table 2, we observe that as the rate λ_h increases, the pick value of $\lambda_l^*(n^*)$ and the number $N=N(\lambda_h)$ at which the latter pick value is attained are both monotonically decreasing, as expected. From Table 1, we also observe that given N, the pick value of $\lambda_l^*(n^*)$ decreases monotonically with increasing rate λ_h, as expected. From Table 2, we observe that the expected delays of the high priority packets and those of the low priority traffic when no miniflags exist, are generally within similar value ranges; the important difference, however, is that an upper bound on the delays of the high priority packets is guaranteed, while with significant probability the delays of the low priority packets can reach large values. We also observe that the existence of miniflags can present a highly significant delay advantage for the low priority traffic. This is clear from the cases where n^* is different than N, (those are the cases where superframes and frames are not identical entities), in Table 2. In the case when $\lambda_h = 0.100$ and N= 7, for example, the absence of miniflags results in expected per low priority packet delays that are generally ten times those induced when miniflags are feasible, (note that $D_\lambda^{B,u}(N,n^*)$ is only an upper bound on expected delays).

Let us now focus on Approach 2. Since the rate of the high priority packets is generally low, as compared to that of the low priority traffic, we selected the λ_h values 0.005, 0.010, 0.050, and 0.100. Regarding the number of the users who may generate high priority packets, we selected values N= 2,3,4,5,6,7; thus, the possibilities for the n values are 1,2,3,4,5,6, and 7. For the above ranges of λ_h and n values, we have that $n^* = \min(n', 2(\lambda_h^{-1} - 1)) = n'$, for any λ_h and n' selection. For every (λ_h, n') pair as above, we computed the value $p^* = 2^{-1}(n^* + 2)\lambda_h$. In addition, for each (N,n') pair as above, we computed the maximum delay, $D_N^{max} \triangleq 2^N(n'+2)$, that a high priority packet may experience. Those results are listed in Table 3. We note that given N, the minimum value of the maximum delay per high priority packet is 3.2^N, and is attained for n=1; thus, the number of users who may generate high priority

traffic determines a lower bound on the feasible maximum per high priority packet delays. For given N and various p values, we found the optimal n values which maximize the throughput $\lambda_l^*(n,p)$ of the low priority traffic, when the only constraint on the per high priority packet delay is the maximum upper bound $2^N(N+2)$. Those results are included in Table 4, together with the corresponding maximum throughput values λ_l^* (n,p), for N = 2,3,4,5,6,7. Next, we considered the case where an upper bound, denoted D^{max}, on the per high priority packet delay is given, which is generally less than the maximum such bound $2^N(N+2)$, for given N. For given N and D^{max}, we then computed the optimal n value that maximizes the throughput $\lambda_l^*(n,p)$ of the low priority traffic, for various values of the probability p. Those results are listed in Table 5, for D^{max} = 25, 50, 75, 100, 150 and N = 2, 3, 4, 5, together with the corresponding maximum throughput values λ_l^* (n,p); as can be seen from Table 3, for N > 5 there is no n value that can attain $D^{max} \leq 150$. In Table 6, we list expected per packet delays for both the high and the low priority packets, for N and n values as those in Table 5, and for Poisson rates λ within the corresponding stability regions which are signified by the respective λ_l^* (n,p) values in Table 5. We note that in both Tables 5 and 6, the values listed for each given n correspond to the range of p values which maintain the rate λ_h of the high priority traffic less than or equal to 0.1. We also note that in Table 6, for the case when "miniflags" are present (case B), the upper bound $D_\lambda^{B,u}$ (N,n,p) of the per low priority packet expected delay is listed, rather than the expected delay itself. In Figures 6, 7, 8, and 9, we plot the expected delays D_λ^A (N, n, p) and the bound $D_\lambda^{B,u}$ (N,n,p) against λ, for the N and n values in Table 6, and for $p \leq 0.20$; that is, for each pair (N,n) we plot the highest corresponding values from Table 6, or equivalently the worst case values for the region $p \leq 0.20$, named then $D_\lambda^A(N,n)$ and $D_\lambda^{B,u}(N,n)$, respectively. In the same figures, we also draw the horizontal lines which correspond to the "worst case" expected per high priority packet delays D_h (N,n,p) in the region $\lambda_h \leq 0.10$; that is, we draw the horizontal lines at the levels D_h (N, n, p ($\lambda_h \leq 0.1$)), for the corresponding (N,n) values. We also indicate the corresponding imposed maximum per high priority packet delay, D^{max}, on each of the curves in the figures. Figures 6, 7, 8, and 9, correspond to N=2, N=3, N=4, and N=5, respectively.

From Figures 6, 7, 8, and 9, we observe the following: (1) Given N, as the imposed maximum per high priority packet delay, D^{max}, increases, so do the delay D_λ^A (N,n) and the bound $D_\lambda^{B,u}$ (N,n) that correspond to the n values which then maximimize the throughput of the low priority traffic, (subject to the D^{max} constraint throughput maximization). This delay penalty is at the gain in throughput of the low priority traffic, which increases as D^{max} increases (see Table 5); we refer to the highest such throughput for given N and D^{max}, obtained at the corresponding optimal n selection (see Table 5). (2) Given N and $D^{max} < 2^N(N+2)$, given then the value n which maximizes the throughput of the low priority traffic (Table 5), the difference $D_\lambda^A(N,n) - D_\lambda^{B,u}(N,n)$ is generally uniformly substantial (i.e., for all λ values within the corresponding stability region). We thus conclude that whenever some $D^{max} < 2^N(N+2)$ is imposed on the high priority packets, then the existence of miniflags presents a significant advantage regarding delays of the low priority traffic, as opposed to the absence of miniflags and presence of flags only. In fact, since $D_\lambda^{B,u}$ (N,n) is generally loose upper bound on the expected per low priority packet delays when miniflags exist, the above advantage is quite more significant than what Figures 6, 7, 8, 9 show, and it is present even when $D^{max} = 2^N(N+2)$, for given N. To take advantage of the delay improvement induced by the existence of miniflags, the system must basically pay in terms of channel capacity dedicated to the miniflag encoding, however. (3) Given N and $D^{max} = \alpha$, let n (N,α) denote the n value which then maximizes the throughput of the low priority traffic. Let also $D_h(N,n,p(\lambda_h \leq 0.1))$ denote then the "worst case," in the region $\lambda_h \leq 0.1$, expected per high priority packet delay, where $n = n(N,\alpha)$. For n = n (N,α), where

$D^{max} = \alpha$, we study the difference $D^{A}_{\lambda}(N,n) - D_h(N,n,p(\lambda_h \le 0.1))$ and $D^{B,u}_{\lambda}(N,n) - D_h(N,n,p(\lambda_h \le 0.1))$. From Figures 6, 7, 8, 9, we observe that given N, the percentage of λ values for which the first of the above two differences is positive increases monotonically with decreasing α value. On the other hand, for α strictly less than $2^N(N+2)$, the second of the above two differences remains negative for almost all the λ values within the corresponding regions, becoming positive and asymptotically large only for λ values close to the respective throughput value, (λ^{*}_{l} (n,p) from Table 5, for $\lambda_h \le 0.1$). The above observations present another strong argument in favor of "miniflags" encoding. Indeed, as D^{max} decreases, the presence of miniflags allows for expected per low priority packet delays that are significantly lower than the expected per high priority packet delays, for all Poisson intensities λ that are not very close to the corresponding throughput of the low priority traffic, while a strict upper bound D^{max} on the per high priority packet delay is simultaneously maintained.

We conclude this section by pointing out that our schemes focus on the effective accommodation of the high priority packets, subject to strict constraints on their maximum delays and system stability, at the expense of increased delays of the low priority traffic. The penalty paid regarding the latter delays is evident from the comparison of the expected delays in Table A of the Appendix, with those in Table 6 and Figures 6, 7, 8, 9, where the delays in Table A correspond to the case when the traffic in the system is uniform and is all low priority. This penalty is significantly less severe when the system provides for the identification of frames, via miniflags.

VI. CONCLUSIONS

We have considered a system where a well-defined finite population of users may generate high priority data, and where an ill-defined, possibly asymptotically large, population of users generate low priority data. We have assumed a strict upper bound on the maximum per high priority packet delay, and we have imposed a stability requirement for the whole system; that is, we do not allow packet losses. For the above system, we proposed and analyzed a transmission algorithm which is a mixture between a deterministic tree search, for the high priority data, and random access, for the low priority packets. The algorithm is limited sensing and is based on the assumption that binary, collision versus noncollision, feedback per slot exists. The algorithm can attain low expected per low priority rpacket delays and relatively high throughputs for the low priority traffic, while it simutaneously maintains a strict upper bound on the maximum per high priority packet delay. The above property becomes stronger as a higher percentage of the channel capacity is dedicated to encoding for the identification of frames. In our quantitative studies, we have ignored the latter percentage, which is generally small. We emphasize that the necessity for the identification of superframes or/and frames, and thus for some waste in channel capacity, is due to the limited sensing environment we considered. When full sensing is possible, (that is, when all users have knowledge of the overall feedback history in the system), then neither flags nor miniflags are necessary, since each user can then identify the starting points of frames without any feedback signaling. Such signaling is the penalty paid by the system for increased flexibility and reliability. Indeed, limited sensing allows for the accommodation of new users, and guarantees stability even when users are temporarily isolated from exposure to system feedbacks, either due to user mobility or due to occasional system failures.

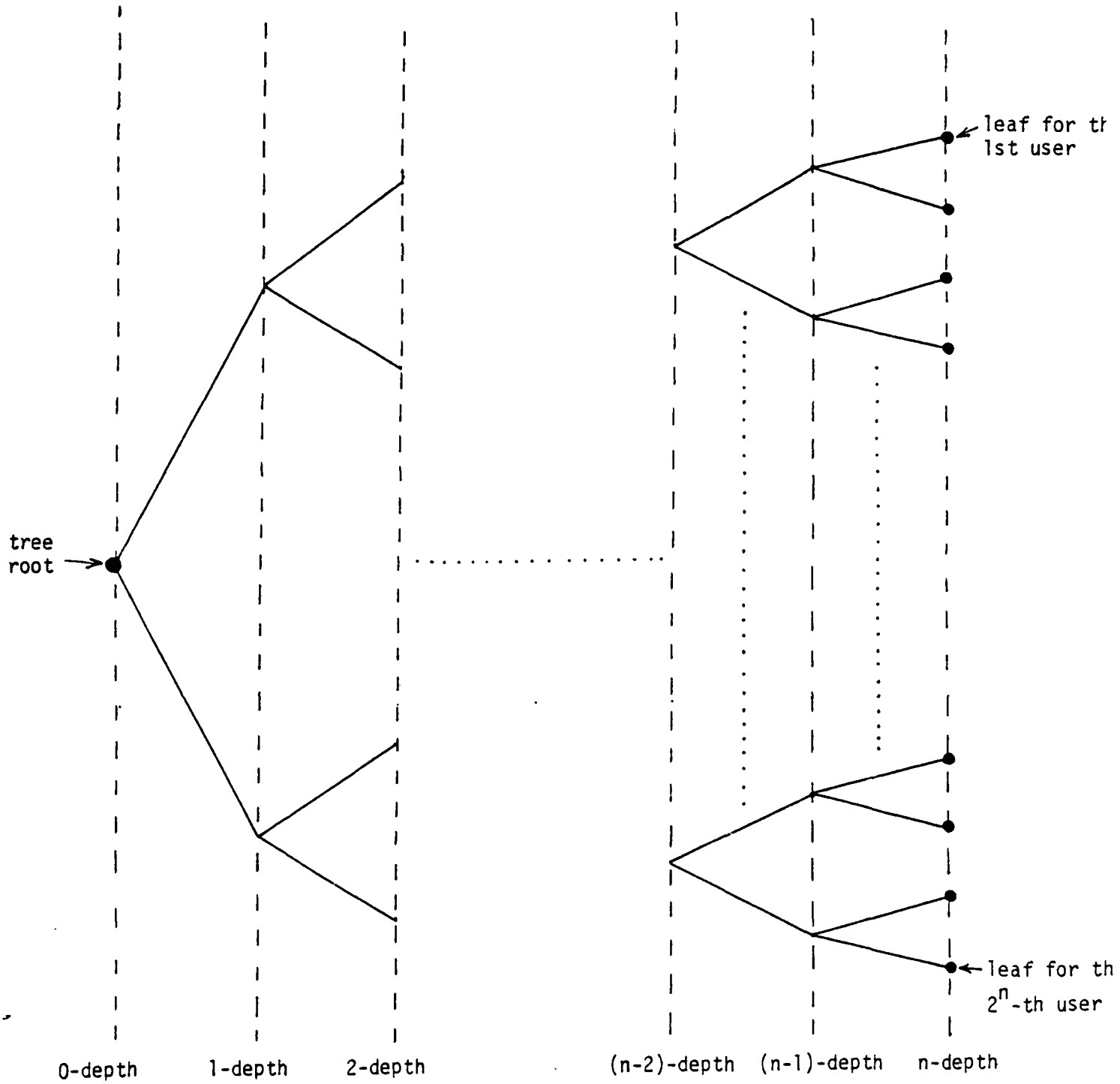


Figure 1

The 2^n-leaves binary tree

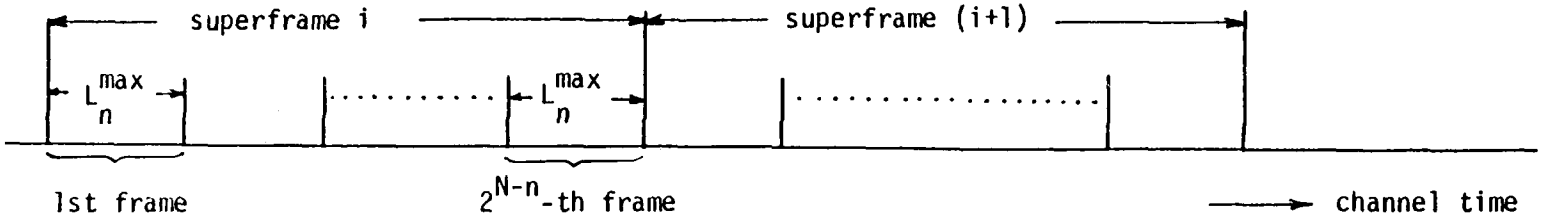

superframe i
superframe (i+1)
L_n^{max}
L_n^{max}
1st frame
2^{N-n}-th frame
channel time

Figure 2

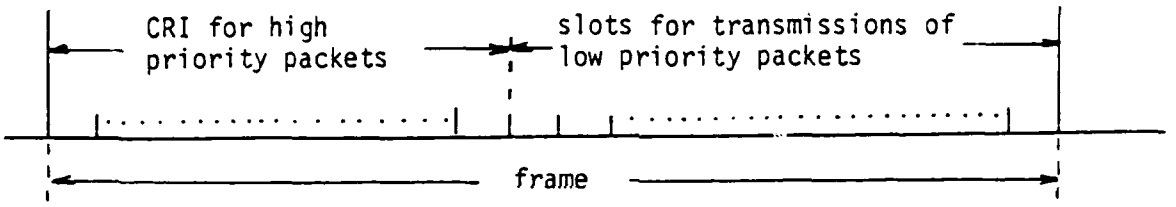
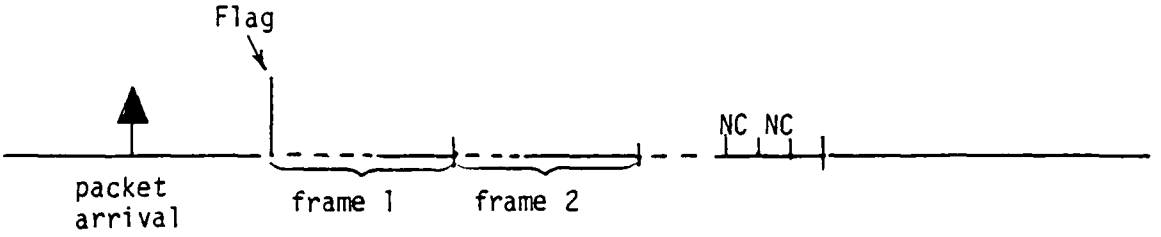

CRI for high
priority packets
slots for transmissions of
low priority packets
frame

Figure 3

Figure 4

The 2-cell algorithm in the limited sensing environment--examined intervals



- - - - : lengths of CRIs for high priority packets, subtracted from the arrival axis

Figure 5

Case A

λ_h	N	p	n	L(n,p)	$D_h(N,n,p)$	$\lambda_l^*(n,p)$
0.020	2	0.040	1	1.0032	4.6016	0.2862
			2*	1.0112	5.3262	0.3756*
0.020	3	0.050	1	1.0050	8.6325	0.2859
			2	1.0175	9.9184	0.3753
			3*	1.6884	13.2201	0.4068*
0.020	4	0.060	1	1.0072	16.6752	0.2856
			2	1.0252	19.0317	0.3749
			3	1.6965	24.3222	0.4061
			4*	3.0981	31.3912	0.4168*
0.020	5	0.070	1	1.0098	32.7485	0.2853
			2	1.0343	37.1988	0.3744
			3	1.7111	46.2974	0.4053
			4	3.1685	57.6940	0.4154
			5*	6.6245	74.1711	0.4173*
0.020	6	0.080	1	1.0128	64.8984	0.2848
			2	1.0448	73.5003	0.3738
			3	1.7321	90.0527	0.4043
			4	3.2619	109.7313	0.4139
			5*	7.0051	135.6280	0.4151*
			6	16.1668	175.1246	0.4134
0.020	7	0.090	1	1.0162	129.2353	0.2843
			2	1.0567	146.1294	0.3732
			3	1.7594	177.4477	0.4032
			4	3.3771	213.3955	0.4121
			5*	7.4390	257.2489	0.4127*
			6	17.4363	316.6386	0.4106
			7	40.6017	408.4741	0.4085
0.020	8	0.100	1	1.0200	258.0200	0.2838
			2	1.0700	291.5425	0.3725
			3	1.7928	352.3047	0.4020
			4*	3.5127	420.6655	0.4102*
			5	7.9201	499.7816	0.4102
			6	18.7875	596.8585	0.4077
			7	43.9580	729.3150	0.4054
			8	100. 7159	939.7010	0.4036
0.050	2	0.100	1	1.0200	4.7600	0.2838
			2*	1.0700	5.8375	0.3725*
0.050	3	0.125	1	1.0312	8.8594	0.2822
			2	1.1094	10.6123	0.3704
			3*	1.9023	15.1875	0.3986*
0.050	4	0.150	1	1.0450	17.0325	0.2802
			2	1.1575	20.0178	0.3678
			3	2.0475	26.8941	0.3944
			4*	4.4576	38.1210	0.3989*
0.050	5	0.175	1	1.0612	33.3969	0.2779
			2	1.2144	38.7594	0.3647
			3	2.2268	49.9034	0.3896
			4*	5.0716	66.5827	0.3922*
			5	12.6192	94.7424	0.3883

λ_h	N	p	n	L(n,p)	$D_h(N,n,p)$	$\lambda_l(n,p)$
0.050	6	0.200	1	1.0800	66.2400	0.2752
			2	1.2800	76.3400	0.3612
			3	2.4384	95.7984	0.3842
			4*	5.7638	122.6222	0.3850*
			5	14.4953	162.7922	0.3801
			6	35.3778	230.7121	0.3756
0.050	8	0.250	1	1.1250	265.0625	0.2688
			2	1.4375	304.4922	0.3527
			3*	2.9521	374.4410	0.3716*
			4	7.3453	460.9123	0.3691
			5	18.5813	566.0403	0.3629
			6	45.1603	704.9174	0.3579
			7	106.3207	919.9693	0.3539
			8	244.6413	1308.6533	0.3508
0.005	2	0.0100	1	1.0002	4.5251	0.2866
			2*	1.0007	5.0804	0.3762*
0.005	3	0.0125	1	1.0003	8.5317	0.2866
			2	1.0011	9.6012	0.3762
			3*	1.7182	12.2967	0.4084*
0.005	4	0.0150	1	1.0004	16.5391	0.2866
			2	1.0016	18.6233	0.3762
			3	1.7131	23.2134	0.4083
			4*	3.1088	28.3211	0.4208*
0.005	5	0.0175	1	1.0006	32.5483	0.2866
			2	1.0021	36.6487	0.3761
			3	1.7086	44.9799	0.4083
			4	3.0926	54.0293	0.4207
			5*	5.8470	64.3931	0.4255*
0.005	6	0.0200	1	1.0008	64.5624	0.2866
			2	1.0028	72.6826	0.3761
			3	1.7044	88.4430	0.4082
			4	3.0783	105.2637	0.4205
			5	5.8231	123.7388	0.4253
			6*	11.4709	146.0157	0.4268*
0.005	7	0.0225	1	1.0010	128.5881	0.2865
			2	1.0035	144.7369	0.3761
			3	1.7007	175.2937	0.4082
			4	3.0661	207.5417	0.4204
			5	5.8066	241.9758	0.4250
			6	11.5014	280.9453	0.4264
			7*	23.6979	331.1084	0.4264*
0.05	8	0.0250	1	1.0012	256.6419	0.2865
			2	1.0044	288.8403	0.3760
			3	1.6975	348.9098	0.4081
			4	3.0557	411.8937	0.4203
			5	5.7975	477.9998	0.4248
			6*	11.5539	549.7136	0.4261*
			7	24.0090	634.1814	0.4259
			8	50.9513	748.1210	0.4254
0.010	2	0.0200	1	1.0008	4.5504	0.2866
			2*	1.0028	5.1616	0.3761*

λ_h	N	p	n	L(n,p)	$D_h(N,n,p)$	$\lambda_l^*(n,p)$
0.010	3	0.0250	1	1.0012	8.5644	0.2865
			2	1.0044	9.7046	0.3760
			3*	1.6975	12.5992	0.4081*
0.010	4	0.0300	1	1.0018	16.5813	0.2864
			2	1.0063	18.7530	0.3759
			3	1.6922	23.5679	0.4079
			4*	3.0407	29.3098	0.4199*
0.010	5	0.0350	1	1.0024	32.6059	0.2863
			2	1.0086	36.8148	0.3758
			3	1.6887	45.3782	0.4077
			4	3.0331	55.1643	0.4195
			5*	5.8284	67.5173	0.4236*
0.010	6	0.0400	1	1.0032	64.6496	0.2862
			2	1.0112	72.9102	0.3756
			3	1.6869	88.8737	0.4074
			4	3.0326	106.5386	0.4191
			5	5.8813	127.3819	0.4229
			6*	12.2527	155.6390	0.4233*
0.010	8	0.0500	2	1.0175	289.4609	0.3753
			3	1.6884	349.3907	0.4068
			4	3.0523	413.5764	0.4181
			5*	6.0533	483.5075	0.4212*
			6	13.0058	564.0397	0.4211
			7	28.9494	666.3399	0.4199
			8	64.2794	814.8743	0.4187
0.10	2	0.200	1	1.0800	5.0400	0.2752
			2*	1.2800	6.7400	0.3612*
0.10	3	0.250	1	1.1250	9.3125	0.2688
			2	1.4375	11.9297	0.3527
			3*	2.9521	18.6827	0.3716*
0.10	4	0.300	1	1.1800	17.8800	0.2609
			2	1.6300	22.1375	0.3424
			3*	3.5768	31.9024	0.3570*
			4	9.1438	49.6262	0.3517
0.10	5	0.350	1	1.2450	35.2125	0.2515
			2	1.8575	42.6841	0.3302
			3*	4.3014	57.9997	0.3404*
			4	11.1204	82.9334	0.3330
			5	27.8230	127.5224	0.3254
0.10	6	0.400	1	1.3200	70.4600	0.2408
			2	2.1200	84.5800	0.3160
			3*	5.1164	110.8828	0.3221*
			4	13.2494	148.9413	0.3132
			5	32.8928	209.1090	0.3053
			6	78.5856	320.0718	0.2994

λ_h	N	p	n	L(n,p)	$D_h(N,n,p)$	$\lambda_l^*(n,p)$
0.10	7	0.450	1	1.4050	142.3825	0.2286
			2	2.4175	170.6534	0.3001
			3*	6.0132	219.4907	0.3022*
			4	15.5146	283.0151	0.2923
			5	38.2274	371.1061	0.2843
			6	90.8548	515.8460	0.2784
			7	210.5097	792.4396	0.2737
0.10	8	0.500	1	1.5000	289.5000	0.2150
			2*	2.7500	348.3125	0.2822*
			3	6.9844	444.0508	0.2808
			4	17.9053	558.6125	0.2705
			5	43.8101	700.1870	0.2625
			6	103.6201	904.2888	0.2566
			7	239.2402	1254.3975	0.2520

For each N and λ_h, the optimal n and the corresponding highest throughput for the low priority class, have a * next to them.

Table 1

Approach 1: Expected per packet delays for the high priority class. Throughput for the low priority class.

λ_h	N	n^*	$D_h(N,n^*)$	$\lambda_l^*(n^*)$	λ	$D_\lambda^\lambda(N,n^*)$	$D_\lambda^{\mu,\upsilon}(N,n^*)$
0.005	2	2	5.0804	0.3762	0.01	6.284	6.284
					0.10	6.626	6.626
					0.15	7.083	7.083
					0.20	7.882	7.882
					0.25	8.910	8.910
					0.30	11.194	11.194
					0.35	14.392	14.392
0.005	6	6	146.0157	0.4268*	0.01	130.092	130.092
					0.10	130.405	130.405
					0.15	130.824	130.824
					0.20	131.556	131.556
					0.25	132.497	132.497
					0.30	134.589	134.589
					0.35	137.518	137.518
0.010	4	4	29.3098	0.4199	0.01	26.132	26.132
					0.10	26.451	26.451
					0.15	26.878	26.878
					0.20	27.624	27.624
					0.25	28.583	28.583
					0.30	30.715	30.715
					0.35	33.700	33.700
0.010	5	5	67.5173	0.4236*	0.01	58.108	58.108
					0.10	58.424	58.424
					0.15	58.845	58.845
					0.20	59.583	59.583
					0.25	60.531	60.532
					0.30	62.640	62.640
					0.35	65.591	65.591
0.020	5	5	74.1771	0.4173*	0.01	58.124	58.124
					0.10	58.442	58.442
					0.15	58.867	58.867
					0.20	59.610	59.610
					0.25	60.566	60.566
					0.30	62.690	62.690
					0.35	65.664	65.664
0.020	7	5	257.2489	0.4127	0.01	226.142	58.142
					0.10	226.463	58.463
					0.15	226.891	58.891
					0.20	227.641	59.641
					0.25	228.605	60.605
					0.30	230.746	62.747
					0.35	233.746	67.746
0.050	4	4	38.1210	0.3989*	0.01	26.204	26.204
					0.10	26.534	26.534
					0.15	26.975	26.975
					0.20	27.746	27.746
					0.25	28.738	28.738
					0.30	30.942	30.942
					0.35	34.028	34.028
0.050	6	4	122.6222	0.3850	0.01	98.272	26.272
					0.10	98.612	26.612
					0.15	99.067	27.067
					0.20	99.862	27.862
					0.25	100.884	28.884
					0.30	103.156	31.156
					0.35	106.557	34.557

λ_h	N	n^*	$D_h(N,n^*)$	$\lambda_l^*(n^*)$	λ	$D_\lambda^A(N,n^*)$	$D_\lambda^{B,u}(N,n^*)$
0.050	8	3	374.4410	0.3716	0.01	322.340	12.340
					0.10	322.691	12.691
					0.15	323.159	13.159
					0.20	323.978	13.978
					0.25	325.031	15.031
					0.30	327.371	17.371
					0.35	330.647	20.647
0.100	3	3	18.6827	0.3716*	0.01	12.346	12.346
					0.10	12.697	12.697
					0.15	13.167	13.167
					0.20	13.988	13.988
					0.25	15.043	15.043
					0.30	17.389	17.389
					0.35	20.674	20.674
0.100	6	3	110.8828	0.3221	0.01	82.686	12.686
					0.10	83.088	13.088
					0.15	83.616	13.616
					0.20	84.566	14.566
					0.25	85.774	15.774
					0.30	88.460	18.460
					0.35	92.221	22.221
0.100	7	3	219.4907	0.3022	0.01	162.858	12.858
					0.10	163.286	13.286
					0.15	163.850	13.850
					0.20	164.858	14.858
					0.25	166.144	16.144
					0.30	169.002	19.002
					0.35	173.003	23.003

Table 2

Approach 1: Expected per packet delays for the low priority class. Cases A and B.

			$D_N^{max} = 2^N(n'+2)$					
λ_h	$n' = n^*$	p^*	N=2	N=3	N=4	N=5	N=6	N=7
0.005	1	0.0075	12	24	48	96	192	384
	2	0.0100	16	32	64	128	256	512
	3	0.0125	-	40	80	160	320	640
	4	0.0150	-	-	96	192	384	768
	5	0.0175	-	-	-	224	448	896
	6	0.0200	-	-	-	-	512	1024
	7	0.0225	-	-	-	-	-	1152
0.010	1	0.0150						
	2	0.0200						
	3	0.0250						
	4	0.0300						
	5	0.0350						
	6	0.0400						
	7	0.0450						
0.050	1	0.0750						
	2	0.1000						
	3	0.1250						
	4	0.1500						
	5	0.1750						
	6	0.2000						
	7	0.2250						
0.100	1	0.1500						
	2	0.2000						
	3	0.2500						
	4	0.3000						
	5	0.3500						
	6	0.4000						
	7	0.4500						

Table 3

Approach 2: Maximum delays of high priority packets, and "worst case" probabilitites p^*.

	N=2		N=3		N=4		N=5		N=6		N=7	
p	n	$\lambda_l^*(n,p)$	n	$\lambda_l^*(n,p)$	n	$\lambda_l^*(n,p)$	n	$\lambda_l^*(n,p)$	n	$\lambda_l^*(n,p)$	n	$\lambda_l^*(n,p)$
0.0075	2	0.3762	3	0.3938	4	0.4016	5	0.4069	6	0.4103	7	0.4127
0.0100	2	0.3762	3	0.3938	4	0.4016	5	0.4071	6	0.4105	7	0.4126
0.0150	2	0.3762	3	0.3938	4	0.4022	5	0.4074	6	0.4107	7	0.4126
0.0200	2	0.3762	3	0.3938	4	0.4024	5	0.4076	6	0.4107	7	0.4125
0.0300	2	0.3759	3	0.3937	4	0.4026	5	0.4077	6	0.4103	7	0.4115
0.0400	2	0.3756	3	0.3936	4	0.4026	5	0.4077	6	0.4094	7	0.4101
0.0500	2	0.3753	3	0.3837	4	0.4027	5	0.4068	6	0.4082	7	0.4084
0.1000	2	0.3725	3	0.3810	4	0.3985	5	0.3996	5	0.3996	5	0.3996
0.2000	2	0.3612	3	0.3776	4	0.3784	4	0.3784	4	0.3784	4	0.3784
0.3000	-	-	-	-	3	0.3531	3	0.3531	3	0.3531	3	0.3531
0.3500	-	-	-	-	-	-	3	0.3375	3	0.3375	3	0.3375
0.4000	-	-	-	-	-	-	-	-	3	0.3200	3	0.3200

Table 4

Approach 2: Optimal n values and optimal throughputs of the low priority traffic, disregarding specific constraints on the maximum delay per high priority packet.

	D^{max}=25				D^{max}=50				D^{max}=75		D^{max}=100				D^{max}=150	
	N=2		N=3		N=3		N=4		N=4		N=4		N=5		N=5	
p	n	$\lambda_l^*(n,p)$	n	$\lambda_l^*(n,p)$	n	$\lambda_l^*(n,p)$	n	$\lambda_l^*(n,p)$	n	$\lambda_l^*(n,p)$	n	$\lambda_l^*(n,p)$	n	$\lambda_l^*(n,p)$	n	$\lambda_l^*(n,p)$
0.0075	2	0.3762	1	0.2867	3	0.3938	1	0.2867	2	0.3762	4	0.4016	1	0.2867	2	0.3762
0.0100	2	0.3762	1	0.2866	3	0.3938	1	0.2866	2	0.3762	4	0.4016	1	0.2866	2	0.3762
0.0200	2	0.3762	1	0.2866	3	0.3938	1	0.2866	2	0.3762	4	0.4022	1	0.2866	2	0.3762
0.0300	2	0.3759	1	0.2864	3	0.3937	1	0.2864	2	0.3759	4	0.4024	1	0.2864	2	0.3759
0.0400	2	0.3756	1	0.2862	3	0.3936	1	0.2862	2	0.3756	4	0.4026	1	0.2862	2	0.3756
0.0500	2	0.3753	1	0.2860	3	0.3837	1	0.2860	2	0.3753	4	0.4027	1	0.2860	2	0.3753
0.1000	2	0.3725	1	0.2838	3	0.3810	1	0.2838	2	0.3725	4	0.3985	1	0.2838	2	0.3725
0.2000	2	0.3612	-	-	3	0.3776	-	-	2	0.3612	4	0.3784	-	-	2	0.3612
0.3000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0.3500	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Note:(1) For $D^{max} \geq 16$: For N=2, n and $\lambda_l^*(n,p)$ values as in D^{max}=25.

(2) For $D^{max} \geq 40$: For N=3, n and $\lambda_l^*(n,p)$ values as in D^{max}=50.

(3) For $D^{max} \geq 96$: For N=4, n and $\lambda_l^*(n,p)$ values as in D^{max}=100.

Table 5

Approach 2: Optimal n values and optimal throughputs of the low priority traffic, subject to constraints on the maximum delay per high priority packet.

λ	p≤0.20							p≤0.05			p=0.10			p=0.20		
	$D_\lambda^{\beta,u}(N,1,p)$	$D_\lambda^{\wedge}(N,1,p)$			$D_\lambda^{\beta,u}(N,2,p)$			$D_\lambda^{\wedge}(N,2,p)$			$D_\lambda^{\wedge}(N,2,p)$			$D_\lambda^{\wedge}(N,2,p)$		
		N=3	N=4	N=5	p≤0.05	p=0.10	p=020	N=2	N=4	N=5	N=2	N=4	N=5	N=2	N=4	N=5
0.01	4.5	9	15	27	6.28	6.30	6.38	6.28	18.28	34.28	6.30	18.30	34.30	6.38	18.38	34.38
0.10	4.95	9.45	15.45	27.45	6.62	6.64	6.73	6.62	18.62	34.62	6.64	18.64	34.64	6.73	18.73	34.73
0.15	5.55	10.05	16.05	28.05	7.07	7.10	7.21	7.07	19.07	35.07	7.10	19.10	35.10	7.21	19.21	35.21
0.20	6.60	11.10	17.10	29.10	7.87	7.91	8.04	7.87	19.87	35.87	7.91	19.91	35.91	8.04	20.04	36.04
0.25	7.95	12.45	18.45	30.45	8.90	8.94	9.11	8.90	20.90	36.90	8.94	20.94	36.94	9.11	21.11	37.11
0.30	-	-	-	-	11.18	11.24	11.49	11.18	23.18	39.18	11.24	23.24	39.24	11.49	23.49	39.49
0.35	-	-	-	-	14.37	14.46	14.83	14.37	26.37	42.37	14.46	26.46	42.46	14.83	26.83	42.83

λ	$D_\lambda^{\beta,u}(3,3,p)=D_\lambda^{\wedge}(3,3,p)$		$D_\lambda^{\beta,u}(4,4,p)=D_\lambda^{\wedge}(4,4,p)$	
	p≤0.10	p=0.20	p≤0.10	p=0.20
0.01	12.18	12.26	26.14	26.26
0.10	12.50	12.60	26.46	26.60
0.15	12.94	13.05	26.89	27.05
0.20	13.70	13.84	27.64	27.84
0.25	14.68	14.86	28.60	28.86
0.30	16.86	17.12	30.74	31.12
0.35	19.92	20.28	33.73	34.28

p	D_h (N,n,p)							
	N=2	N=3		N=4			N=5	
	n=2	n=1	n=3	n=1	n=2	n=4	n=1	n=2
0.0075	5.06	8.51	11.14	16.51	18.56	25.78	32.51	36.56
0.01	5.08	8.52	11.19	16.52	18.58	25.92	32.52	36.58
0.02	5.16	8.55	11.40	16.55	18.66	26.51	32.55	36.67
0.03	5.24	8.57	11.60	16.58	18.75	27.15	32.58	36.76
0.04	5.32	8.60	11.82	16.61	18.84	27.83	32.62	36.86
0.05	5.40	8.63	12.04	16.64	18.93	28.54	32.66	36.97
0.10	5.83	8.78	13.19	16.82	19.44	32.50	32.90	37.58
0.20	6.74	-	15.76	-	20.66	41.28	-	39.22

Table 6

Approach 2: Expected delays for the high and the low priority traffics

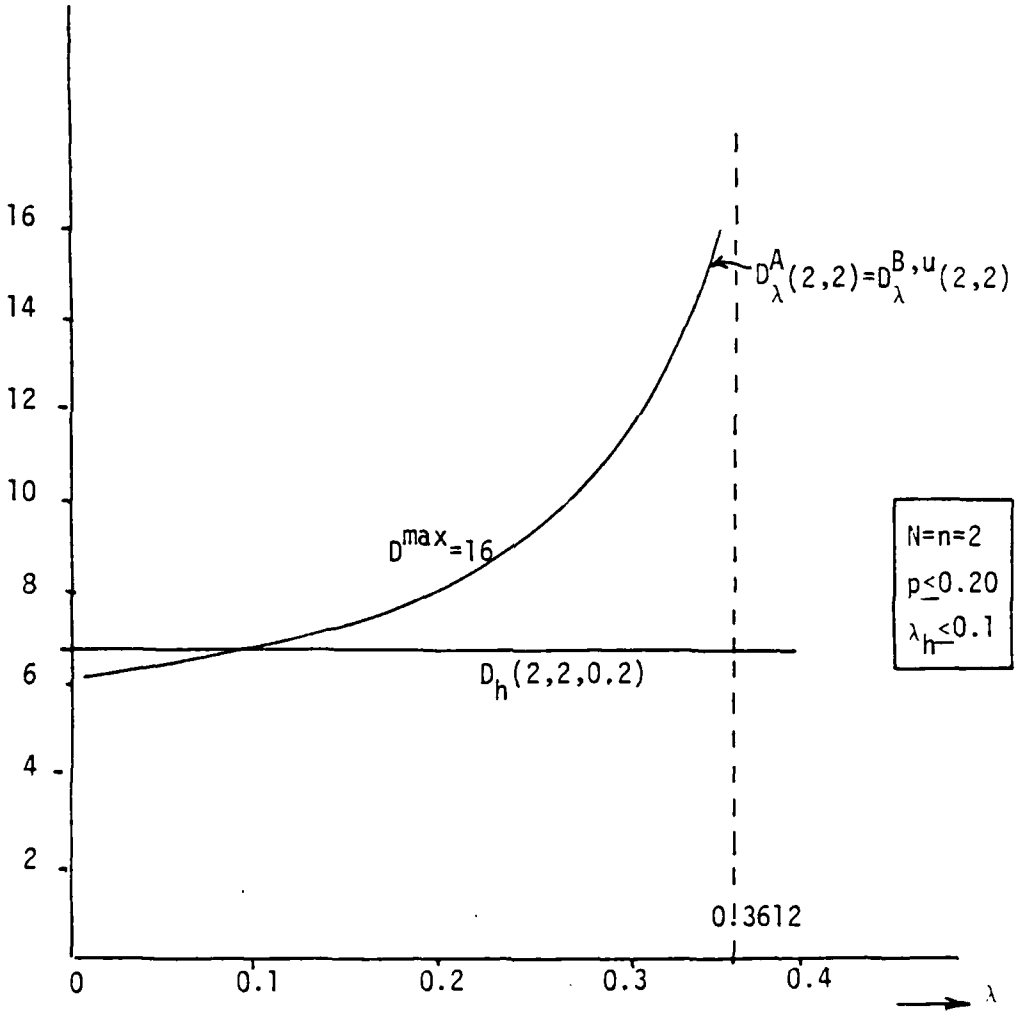


Figure 6

Approach 2: Expected per packet delays

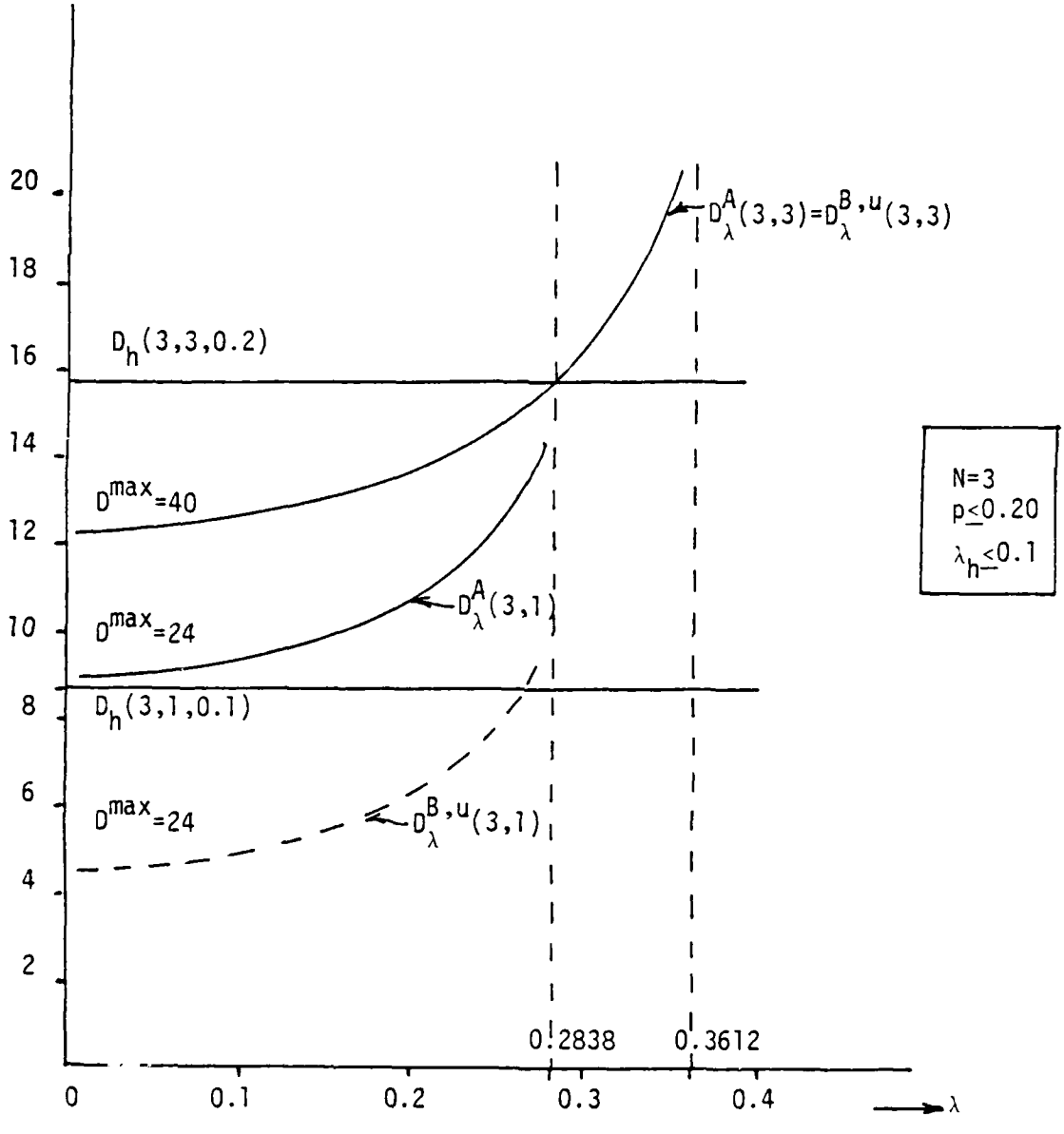


Figure 7

Approach 2: Expected per packet delays

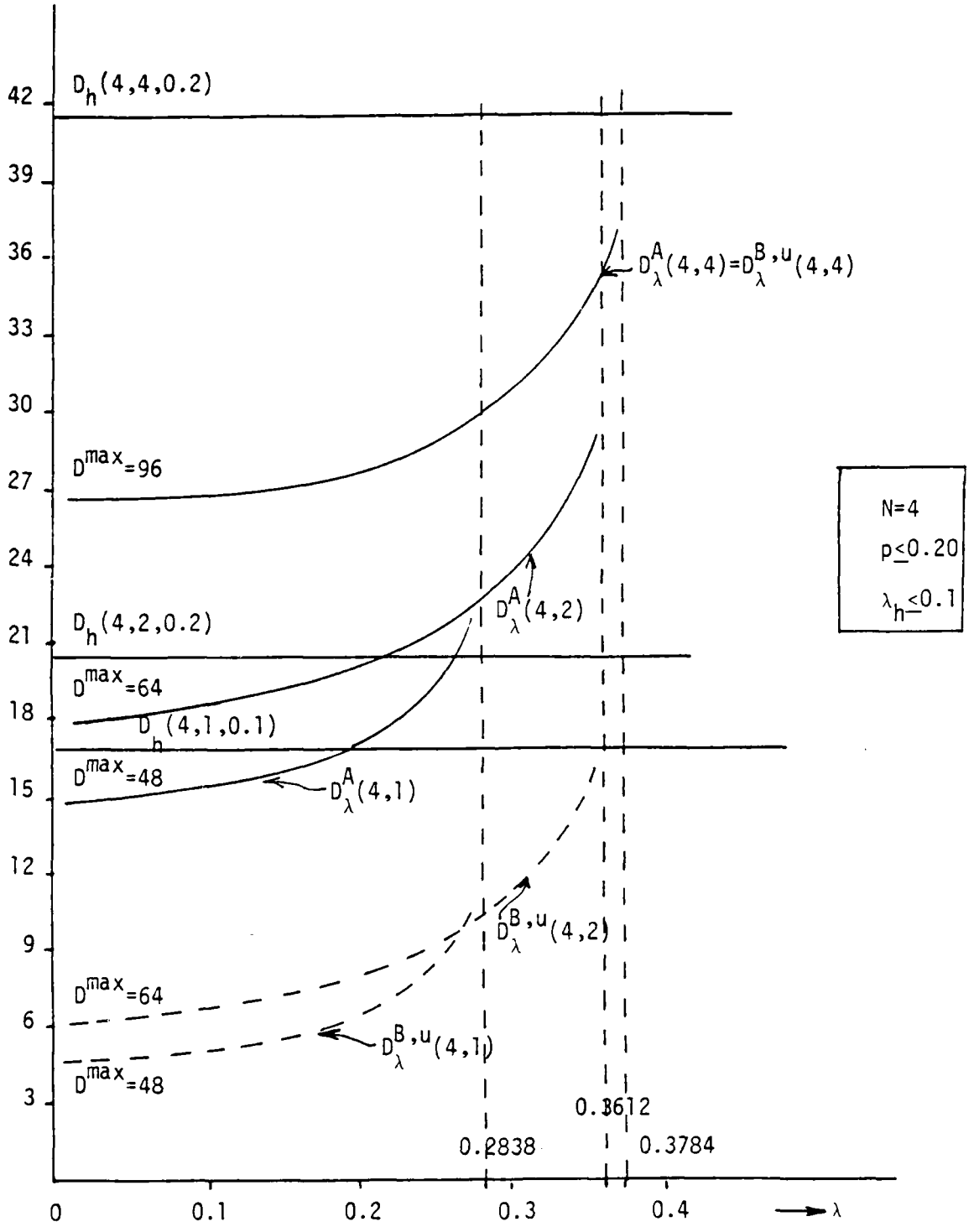


Figure 8

Approach 2: Expected per packet delays

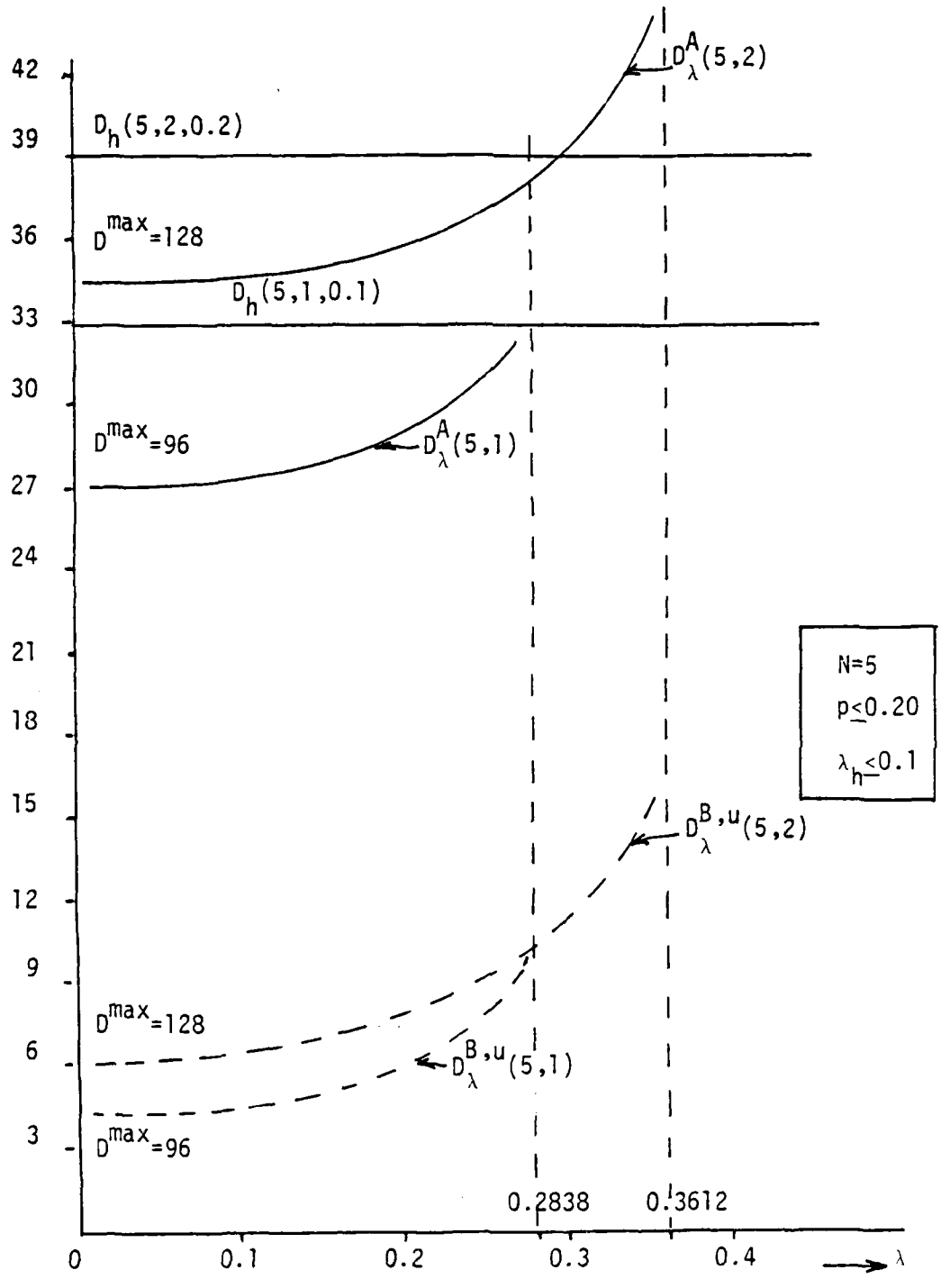


Figure 9

Approach 2: Expected per packet delays

APPENDIX

Proof of Proposition 1

The algorithm clearly induces the following recursive expression:

$$L_n^{max} = 2^{n-1} + 2\, L_{n-1}^{max} \tag{A.1}$$

; where

$$L_1^{max} = 3 \tag{A.2}$$

Expressions (A.1) and (A.2) easily give the expression in (1).

The Evaluation of D_λ

We consider the prototype system defined in Section IV.2, and we focus on the evaluation of the expected per packet delay, D_λ, for given Poisson intensity λ. Given the starting point of a CRI induced by the system, we define as lag at this starting point, the total length of arrival intervals which contain unexamined packet arrivals, where from this length the slot immediately preceeding the starting point of the CRI is excluded. As previously presented in [11], it can be seen that the system generates regenerative points within its stability region. In particular, those regenerative points are the sequence of starting points of CRIs, with lag equal to one. Let us define:

W_λ: The expected cumulative delay of all packets transmitted in the interval between two consecutive regenerative points, when the intensity of the Poisson traffic is λ, (lying within the stability region of the system).

Z_λ: Given intensity λ of the Poisson traffic, (lying within the stability region of the system), the expected number of packets transmitted in the interval between two consecutive regenerative points.

Then, as fully explained in [4], the regenerative theorem gives:

$$D_\lambda = W_\lambda\, Z_\lambda^{-1} \tag{A.3}$$

Let us now define the following quantities:

Ψ_d: Given lag d at the beginning of a CRI, the expected cumulative waiting time of all the packets that will be transmitted in the interval between the beginning of the CRI and the next regenerative point (first time that a CRI with lag one will occur).

Φ_d: Given lag d at the beginning of a CRI, the expected cumulative transmission time of all the packets that will be transmitted in the interval between the beginning of the CRI and the next regenerative point.

w_d: Given lag d at the beginning of a CRI, the expected cumulative transmission time of those packets transmitted during the CRI.

θ_d: Given lag d at the beginning of a CRI, the expected cumulative waiting time of all the packets transmitted during the CRI.

H_d: Given lag d at the beginning of a CRI, the number of slots from the beginning of this CRI to the next CRI beginning with lag one.

$P_x(l)$: Given a CRI which examines an interval of length x, the probability that the CRI will be l slots long.

Relating the quantities in (A.3), with those defined above, we initially have:

$$W_\lambda = \Psi_1 + \Phi_1$$

$$Z_\lambda = \lambda H_1 \tag{A.4}$$

Towards the computation of the quantities in (A.4), we first write the following recursions and expressions that are induced by the algorithmic system:

$$\theta_d = \begin{cases} \lambda d \,(\frac{d}{2} + 1) \;, \text{ if } d \le \Delta \\ \lambda \Delta \,(\frac{\Delta}{2} + 1) \;, \text{ if } d > \Delta \end{cases} \tag{A.5}$$

$$\Psi_d = \begin{cases} \lambda d \,(\frac{d}{2} + 1) + \sum_{l=1}^{\infty} \Psi_l \, P_d \,(l) \;, \text{ if } d \le \Delta \\ \lambda \,\Delta(\frac{\Delta}{2} + 1) + \sum_{l=1}^{\infty} \Psi_{d-\Delta+l} \, P_\Delta(l) \;, \text{ if } d > \Delta \end{cases} \tag{A.6}$$

$$\Phi_d = \begin{cases} w_d + \sum_{l=1}^{\infty} \Phi_l \, P_d \,(l) \;, \text{ if } d \le \Delta \\ w_\Delta + \sum_{l=1}^{\infty} \Phi_{d-\Delta+l} \, P_\Delta(l) \;, \text{ if } d > \Delta \end{cases} \tag{A.7}$$

$$H_d = \begin{cases} \sum_{l=1}^{\infty} [l + H_l] \, P_d(l) \;, \text{ if } d \le \Delta \\ \sum_{l=1}^{\infty} [l + H_{d-\Delta+l}] \, P_\Delta(l) \;, \text{ if } d > \Delta \end{cases} \tag{A.8}$$

; where for $d \le \Delta$, $H_d = 1$, with probability $e^{-\lambda d}\,(1+\lambda d)$.

The expressions in (A.6), (A.7), and (A.8) determine linear systems of infinite dimensionality. The methodology in [4] is used for the evaluation of upper and lower bounds on the respective quantities. Towards the evaluation of the expected value w_d, and the probabilities $P_x(l)$ in the above systems, we define:

$Z_{n,k-n}$: Given a time instant within a CRI, such that n packets have counter value equal to 1, and k-n packets have counter value equal to 2, the expected cumulative delay of all the packets transmitted from the point that the event (n,k-n) occurs, to the end of the CRI.

$l_{k,m}$: Given k packets with counter values equal to 1 and m packets with counter values equal to 2, the number of slots needed by the algorithm until the first successful transmission, (and including it), after the k-multiplicity collision has been observed.

$P_k(l)$: Given a k-multiplicity initial collision, the probability that it takes l slots for its resolution, including the initial collision slot.

The operations of the algorithmic system determine then the following expressions:

$$Z_{0,0} = 0,\ Z_{1,0} = 1,\ Z_{0,k} = k + Z_{k,0}\ ;\ k \geq 1$$

$$Z_{1,k-1} = k + Z_{0,k-1} = 2k-1 + Z_{k-1,0}\ ;\ k \geq 2 \tag{A.9}$$

$$Z_{n,k-n} = k + 2^{-n} \sum_{i=0}^{n} \binom{n}{i} Z_{i,k-i}\ ;\ n \geq 2$$

And

$$w_d = \begin{cases} \sum_{k \geq 0} e^{-\lambda d} \frac{(\lambda d)^k}{k!} Z_{k,0}\ ;\ d \leq \Delta \\ \sum_{k \geq 0} e^{-\lambda \Delta} \frac{(\lambda \Delta)^k}{k!} Z_{k,0}\ ;\ d > \Delta \end{cases} \tag{A.10}$$

Also,

$$P(l_{1,0} = 0) = P(l_{0,0} = 0) = 1$$

$$P(l_{0,1} = 1) = P(l_{1,1} = 1) = 1$$

$$P(l_{2,0} = 2) = \frac{1}{2} \tag{11}$$

$$\text{For } s \geq 3;\ \begin{cases} P(l_{0,m} = s) = P(l_{1,m} = s) = P(l_{m,0} = s-1) \\ k \geq 2\ ;\ P(l_{k,m} = s) = 2^{-k} \sum_{i=0}^{k} \binom{k}{i} P(l_{i,k+m-i} = s-1) \end{cases}$$

And,

$$P_0(1) = P_1(1) = 1$$

$$P_2(l) = P(l_{2,0} = l-2)\ ,\ \text{for } l \geq 4 \tag{A.12}$$

$$k \geq 3,\ l \geq k+2\ ;\ P_k(l) = \sum_{s=1}^{l-k-1} P(l_{k,0} = s) P_{k-1}(l-s-1)$$

where,

$$P_x(l) = \sum_{k=0}^{\infty} e^{-\lambda x} \frac{(\lambda x)^k}{k!} P_k(l) \qquad \text{(A.13)}$$

Our methodology provides upper and lower bounds on D_λ which are identical to each other to the first decimal point, for λ values less than on equal to 0.35, (the throughput of the algorithm in the prototype system equals 0.43). Below, we provide the D_λ values, to the first decimal point.

λ	D_λ
0.01	2.5
0.10	2.8
0.15	3.2
0.20	3.9
0.25	4.8
0.30	6.8
0.35	9.6

Table A

REFERENCES

[1] J. I. Capetanakis, "Tree Algorithms for the Packet Broadcast Channel," IEEE Trans. inform. Theory, vol. IT-25, pp. 505-515, Sept. 1979.

[2] J. I. Capetanakis, "Generalized TDMA: The Multi-Accessing Tree Protocol," IEEE Trans. Commun., vol. COM-27, pp. 1476-1484, Oct. 1979.

[3] R. G. Gallager, "Conflict Resolution in Random Access Broadcast Networks," in Proc. AFOSR Workshop Communication Theory and Applications, Provincetown, MA, Sept. 1978, pp. 74-76.

[4] L. Georgiadis, L. Merakos, and P. Papantoni-Kazakos, "A Method for the Delay Analysis of Random Multiple Access Algorithms whose Delay Process is Regenerative," IEEE Journal on Selected AReas in Communications, vol. SAC-5, pp. 1051-1062, July 1987.

[5] L. Georgiadis and P. Papantoni-Kazakos, "A 0.487 Throughput Limited Sensing Algorithm," IEEE Trans. on Information Theory, Vol. IT-33, No. 2, March 1987, pp. 233-237.

[6] P. Humblet, "On the Throughput of Channel Access Algorithms with Limited Sensing," IEEE Trans. on Communications, Vol. COM-34, April 1986, pp 345-347

[7] J. Kurose, M. Schwartz, and Y. Yemini, "Multiple Access Protocols and Time Constrained Communication," ACM Computing Surveys, vol. 16, no. 1, March 1984.

[8] P. Papantoni-Kazakos, M. Paterakis, and Liu Ming, "Interconnection Algorithms in Mobile C^3 Topologies," *1988 C^3 Symposium, proceedings. Also, submitted for journal publication.*

[9] M. Paterakis, L. Georgiadis, and P. Papantoni-Kazakos, "On the Relation between the Finite and the Infinite Population Models for a Class of RAAs," IEEE Trans. Comm., vol. COM-35, pp. 1239-1240, Nov. 1987.

[10] M. Paterakis, L. Georgiadis, and P. Papantoni-Kazakos, "A Full Sensing Window Random-Access Algorithm for Messages with Strict Delay Constaints," Report No. UVA/515415/EE87/103, School of Engineering and Applied Science, University of Virginia, Charlottesville, Virginia, Feb. 1987. Also, Algorithmica, An International Journal in Computer Science, Springer-Verlag, to appear in 1988.

[11] M. Paterakis and P. Papantoni-Kazakos, "A Simple Window Random Access Algorithm with Advantageous Properties," Proceedings of INFOCOM'88. Also, submitted for journal publication.

[12] I. Stavrakakis and D. Kazakos, "A Multi-User Random Access Communication System for Users with Different Priorities," Univ. of Virginia, Technical Report UVA/525415/EE87/104, March 1987. Also, submitted for journl publication.

[13] B. S. Tsybakov, and N. D. Vvedenskaya, "Random Multiple Access Stack Algorithm," Problemy Peredachi Informatsii, vol. 16, no. 3, pp. 80-94, July-Sept. 1980.

DISTRIBUTION LIST

Copy No.	
1 - 6	Director Naval Research Laboratory Washington, DC 20375 Attention: Code 2627
7	Dr. R. N. Madan Electronics Division, Code 1114SE Office of Naval Research 800 N. Quincy Street Arlington, VA 22217-5000
8	Mr. James G. Smith Office of Naval Research Code 1241 800 N. Quincy Street Arlington, VA 22217-5000
9	Professor Mike Athans MIT, Building 35 Cambridge, MA 02139
10	Professor A. Ephremides Electrical Engineering Dept. University of Maryland College Park, MD 20742
11	Dr. Dave Castanon ALPHATECH, Inc. 2 Burlington Executive Center 111 Middlesex Turnpike Burlington, MA 01803-4901
12	Dr. John P. Lehoczky Department of Statistics Carnegie Mellon University Schenley Park Pittsburgh, PA 15213-3890
13	Professor Alex Levis MIT, Building 35 Cambridge, MA 02139
14	Dr. Harold L. Hawkins Manager, Perceptual Sciences Code 1142 Office of Naval Research 800 N. Quincy Street Arlington, VA 22217-5000

15 Dr. G. Malecki
Office of Naval Research
Code 1142
800 N. Quincy Street
Arlington, VA 22217-5000

16 Dr. David Kleinman
ESE Department, U-157
The University of Connecticut
STORRS, CT 06268

17 - 28 Defense Technical Information Center, S47031
Building 5, Cameron Station
Alexandria, VA 22314

29 - 30 P. Papantoni-Kazakos, EE

31 - 32 E. H. Pancake, Clark Hall

33 SEAS Publications Files

* Office of Naval Research Resident Representative
818 Connecticut Avenue, N.W.
Eighth Floor
Washington, DC 20006

Attention: Mr. Michael McCracken
Administrative Contracting Officer

*Send Cover Letter Only

JO#1969:ph